# 

Welcome Aliens!!

Let’s Explore!!!

**Contents**

1. SDLC
2. Web Automation
3. JMeter
4. Appium
5. Programming Language (Core Java)
6. Domain Knowledge
7. General Interview Questions
8. AWS
9. Jenkins
10. Git
11. Various Article

**Note:** Please go to <View><Show document outline> You can see all the contents of the file. This is a collections only, you can read and verify as required as I tried to provide all the references.

# 1 SDLC

## 1.1 Agile

### 1.1.1 Story Point

A story point is an abstract measure of effort required to implement a user story. In simple terms, it is a number that tells the team about the difficulty level of the story. Difficulty could be related to complexities, risks, and efforts involved.

<http://www.tothenew.com/blog/how-to-estimate-story-points-in-agile/>

Story points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work.

**In most cases a story point uses one of the following scales for sizing:**

• 1,2,4,8,16

• X-Small, Small, Medium, Large, Extra-Large ( known as “T-Shirt Sizing”)

XS = 1, S = 2, M = 5, L = 13 , XL = 20, XXL = 40

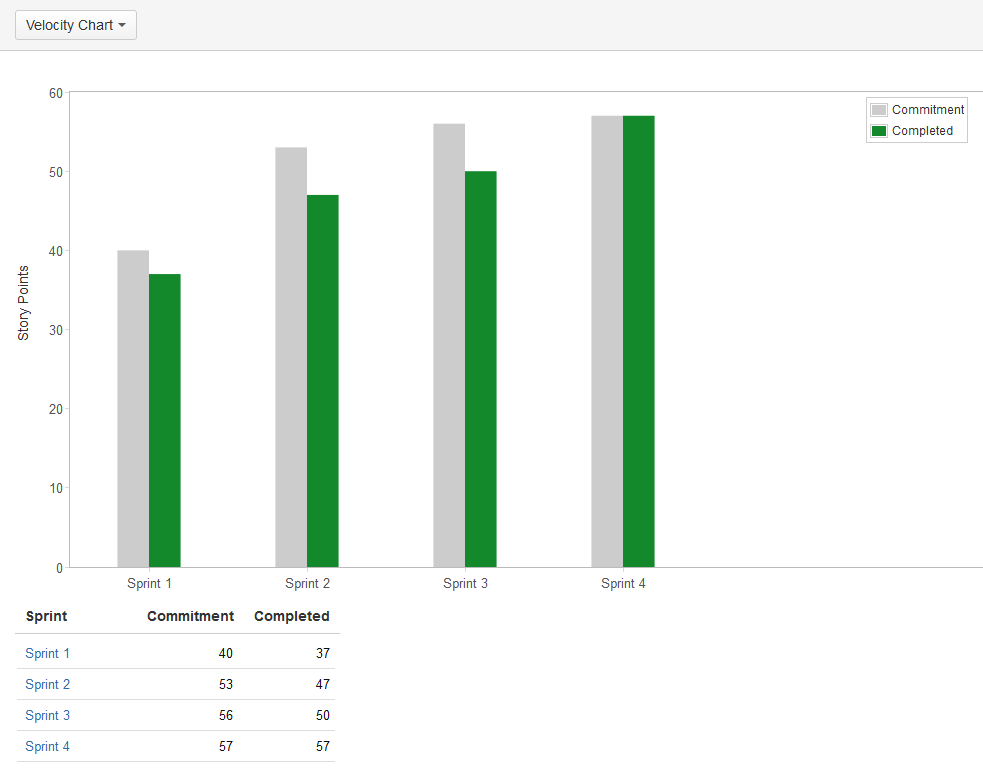
• Fibonacci sequence: 1,2,3,5,8,13,21……...

Scrum Team capacity and velocity calculated based on team's productivity and business value delivered to customer.

### 1.1.2 What is Team's velocity

**Number of story points delivered/demo in a Sprint is called velocity.** For example, if team planned 30 story point(Business value) worth of user stories in a sprint and able to deliver as planned then team's velocity is 30.

The velocity can be estimated as the average, over several recent sprints, of the sum of the estimates for the amount of work completed by a team per sprint — so in the chart above, the velocity = (37 + 47 + 50 +57) / 4 = 48. A team's recent velocity can be useful in helping to predict how much work can be completed by the team in a future sprint.



### 1.1.3 What is Team's capacity?

Total **number of available hours for a sprin**t is called Team's Capacity. Available hours calculated based on resources planned holiday and company holiday if any.

### 1.1.4 Epics

Collection of complex user stories may be grouped as Epics, which can be broken down further.

Each Epic and user story contains acceptance criteria with necessary business rules and mockups attached to them.

### 1.1.5 Scrum

* As per Scrum, no changes are allowed during the sprint by the Product Owner.
* The Product Owner cannot add new stories to the sprint backlog during the sprint and distract the team from the executing the sprint goal.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 1.2 JIRA

### 1.2.1 Sprint Report

The **Sprint Report** shows the list of issues in each **sprint**. It is useful for your **Sprint** Retrospective meeting, and also for mid-**sprint** progress checks. If you have Confluence linked to your JIRA instance, you can also create and/or link Confluence pages to your **sprint report**.

## 1.3 QA estimates

**Introduction:**

Many have searched the World Wide Web (WWW) looking for good software testing estimate templates and have not found one. I did the same thing and then I decided to create my own estimating template. I have been using it for years to create testing estimates, in many different work environments, and have compared estimates to actuals using my template. It is stable and reliable, I have found time after time and have not searched for anything else nor will I. The only thing I ask if you decide to use it, is to give credit where credit is due to Soft-e somewhere in your documentation.

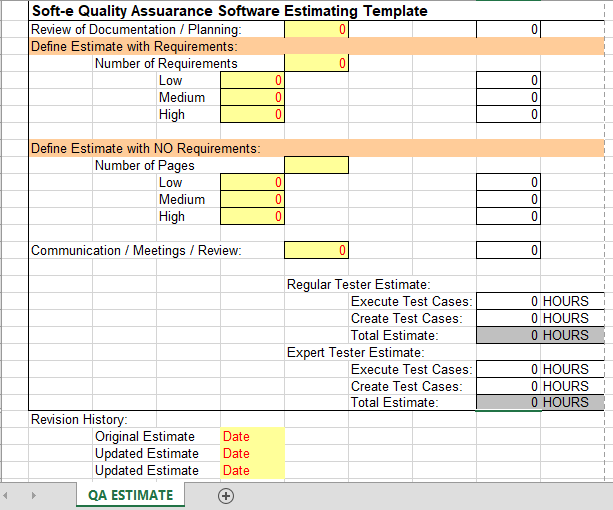
**Why is Estimating Important?**

There are a lot of people that do not believe in estimating or the importance of estimating. It is imperative that estimates be completed to inform early on if the project is at risk based off an estimate. There could already be pre-estimates created by others along the way and creating a testing estimate to compare to these pre-estimates can easily put the project on the right path or on an alert status. Testing estimates are to inform project stake holders if the timeline creation is skewed or if more testing bodies will be needed.

**Quality Assurance Software Estimate:**

Using the spreadsheet attached, the only thing you need to fill out are the yellow fields (which are described below in more detail): Review of Documentation/Planning field, (Pick one section to fill out: Define estimate with or without requirements), Communications/Meetings/Review, and Revision History. The calculations will automatically be performed for you in the Regular/Advanced Tester fields and those are your estimates.

To calculate approximately how long the quality assurance processes will take during a software life-cycle. This estimate can be created with or without requirements. The total estimate includes document processing and review, creating any required documentation, fully testing the software, and accounting for defect management re-testing time.



**Requirement:**

A singular documented need of what a particular product or service should be or perform.

**Test Scenario:**

Defines exact test steps covering all requirements and functions changing.

**Review of Documentation / Planning:**

This is calculated by using a 5 minute time frame multiplied by the number of pages provided to the tester regarding the web site in its entirety rounded to the nearest:

15 minutes = 0.25

30 minutes = 0.50

45 minutes = 0.75

60 minutes = 1.00

For example:

A ten page document would be (10 pages x .05 minutes =0.50 or 30 minutes) and 0.50 is what would be placed in the review of documentation / Planning field.

**Define Estimate with Requirements:**

A requirements document or Use Case is available to the tester.

**Define Estimate without Requirements:**

A requirements document or Use Case is NOT available to the tester.

**High:**

A test scenario is deemed high if it will take a considerable amount of time to test, could be a more complicated software function, or is a difficult database query.

**Medium:**

A test scenario is deemed medium if it has data set-up needs prior to test execution, additional research might be needed, or it requires advanced technical knowledge.

**Low:**

A test scenario is deemed low if it requires basic knowledge of computers for execution, it can be passed off to test if needed by another source, or will not take up a significant amount of execution time.

**Beginner Tester:**

A beginner tester would be a tester one or beginner tester. It could also mean that the application/web site is brand new and everyone is a beginner on the project.

**Expert Tester:**

An expert tester would be an advanced tester or senior tester, whether the application/web site is brand new or not does not apply here.

**Pages:**

With Requirements:

The number of pages that are contained within the Business Requirements Document, Use Case, or any other document(s) that contain requirements.

Without Requirements:

Are development pages for an existing application/web site. Examples are: .CSS .HTML .ASPX .ASP .JAVA

**Communication / Meetings / Review:**

This is defined by a generalization of how big the application/web site is related to the experience of the tester.

**Beginner Tester:**

If a small web site (less than 10 pages) = 4 hours

If a medium web site (11 – 49 pages) = 8 hours

If a large web site (50 or more pages) = 16 hours

**Expert Tester:**

If a small web site (less than 10 pages) = 2 hours

If a medium web site (11 – 49 pages) = 4 hours

If a large web site (50 or more pages) = 8 hours

## 1.4 SDLC Questions

Reference website for bellow 200 SDLC Questions: <http://istqbexamcertification.com>

1. What is a Bug?

When actual result deviates from the expected result while testing a software application or product then it results into a defect. Hence, any deviation from the specification mentioned in the product functional specification document is a defect. In different organizations it’s called a Bug.

2. What is a Defect?

If software misses some feature or function from what is there in requirement it is called as defect.

3. What is CMM?

The Capability Maturity Model for Software (CMM or SW-CMM) is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes.

4. What is Beta Testing?

Beta testing is testing of a release of a software product conducted by customers.

5. What is Black Box Testing?

Testing based on an analysis of the specification of a piece of software without reference to its internal workings. The goal is to test how well the component conforms to the published requirements for the component.

6. What is Bottom Up Testing?

An approach to integration testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

7. What is Boundary Testing?

Test which focus on the boundary or limit conditions of the software being tested. (Some of these tests are stress tests).

8. What is Boundary Value Analysis?

BVA is similar to Equivalence Partitioning but focuses on "corner cases" or values that are usually out of range as defined by the specification. This means that if a function expects all values in range of negative 100 to positive 1000, test inputs would include negative 101 and positive 1001.

9. What is Branch Testing?

Testing of all branches in the program source code is called Branch Testing.

10. What is Coding?

The generation of source code is called Coding.

11. What is Compatibility Testing?

Testing whether software is compatible with other elements of a system with which it should operate, e.g. browsers, Operating Systems, or hardware.

12. What is a Component?

A component is an identifiable part of a larger program or construction. Usually, a component provides a particular function or group of related functions.

13. What is Component Testing?

Testing of individual software components is called Component testing.

14. What is Acceptance Testing?

Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

15. What is Accessibility Testing?

Verifying a product is accessible to the people having disabilities (deaf, blind, mentally disabled etc.).

16. What is Ad Hoc Testing?

A testing phase where the tester tries to 'break' the system by randomly trying the system's functionality is called Ad Hoc testing. This can include negative testing also.

17. What is Agile Testing?

Testing practice for projects using agile methodologies, treating development as the customer of testing and emphasizing a test-first design paradigm. See also Test Driven Development.

18. What is Application Binary Interface (ABI)?

A specification defining requirements for portability of applications in binary forms across different system platforms and environments is called Application Binary Interface (ABI).

19. What is Application Programming Interface (API)?

A formalized set of software calls and routines that can be referenced by an application program in order to access supporting system or network services is called Application Programming Interface (API).

20. What is Automated Software Quality (ASQ)?

The use of software tools, such as automated testing tools, to improve software quality is called Automated Software Quality (ASQ).

21. What is Automated Testing?

Testing employing software tools which execute tests without manual intervention is called Automated Testing. Can be applied in GUI, performance, API, etc. testing. The use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.

22. What is Backus-Naur Form?

It is a meta-language used to formally describe the syntax of a language.

23. What is Basic Block?

A sequence of one or more consecutive, executable statements containing no branches is called Basic Block.

24. What is Basis Path Testing?

A white box test case design technique that uses the algorithmic flow of the program to design tests.

25. What is Basis Set?

The set of tests derived using basis path testing.

26. What is Baseline?

The point at which some deliverable produced during the software engineering process is put under formal change control.

27. What you will do during the first day of job?

Few things that you should be doing on the first day of your job are 1. Reach office at least 15 minutes early

2. Be calm and relaxed

3. Have a smile on your face

4. Don’t be shy

5. Don’t try too hard to impress your colleagues

6. If you're offered to go have lunch with your new boss and co-workers then you should go. It's important to show that you're ready to mingle with your new team.

7. Pay attention to how decisions are made.

28. What is Binary Portability Testing?

Testing an executable application for portability across system platforms and environments, usually for conformation to an ABI specification is called Binary Portability Testing.

29. What is Breadth Testing?

A test suite that exercises the full functionality of a product but does not test features in detail is called Breadth Testing.

30. What is CAST?

Computer Aided Software Testing refers to the computing-based processes, techniques and tools for testing software applications or programs.

31. What is Capture/Replay Tool?

A test tool that records test input as it is sent to the software under test. The input cases stored can then be used to reproduce the test at a later time. Most commonly applied to GUI test tools.

32. What is Cause Effect Graph?

A graphical representation of inputs and the associated outputs effects which can be used to design test cases.

33. What is Code Complete?

Phase of development where functionality is implemented entirety; bug fixes are all that are left. All functions found in the Functional Specifications have been implemented.

34. What is Code Coverage?

An analysis method that determines which parts of the software have been executed (covered) by the test case suite and which parts have not been executed and therefore may require additional attention.

35. What is Code Inspection?

A formal testing technique where the programmer reviews source code with a group who ask questions analyzing the program logic, analyzing the code with respect to a checklist of historically common programming errors, and analyzing its compliance with coding standards. Know more about the Inspection in software testing.

36. What is Code Walkthrough?

A formal testing technique where source code is traced by a group with a small set of test cases, while the state of program variables is manually monitored, to analyze the programmer's logic and assumptions. Know more about Walkthrough in software testing.

37. What is Concurrency Testing?

Multi-user testing geared towards determining the effects of accessing the same application code, module or database records. Identifies and measures the level of locking, deadlocking and use of single-threaded code and locking semaphores.

38. What is Conformance Testing?

Conformance testing, also known as compliance testing, is a methodology used in engineering to ensure that a product, process, computer program or system meets a defined set of standards.

39. What is Context Driven Testing?

The context-driven school of software testing is flavour of Agile Testing that advocates continuous and creative evaluation of testing opportunities in light of the potential information revealed and the value of that information to the organization right now.

40. What is Conversion Testing?

Testing of programs or procedures used to convert data from existing systems for use in replacement systems.

41. What is Cyclomatic Complexity?

A measure of the logical complexity of an algorithm, used in white-box testing is called Cyclomatic Complexity.

42. What is Data Dictionary?

A database that contains definitions of all data items defined during analysis.

43. What is Data Flow Diagram?

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects.

44. What is Data Driven Testing?

Testing in which the action of a test case is parameterized by externally defined data values, maintained as a file or spreadsheet. This is a common technique used in Automated Testing.

45. What is Debugging?

The process of finding and removing the causes of software failures is called Debugging.

46. What is Defect?

When actual result deviates from the expected result while testing a software application or product then it results into a defect.

47. What is Dependency Testing?

Dependency Testing, a testing technique in which an application's requirements are pre-examined for an existing software, initial states in order to test the proper functionality. The impacted areas of the application are also tested when testing the new features or existing features.

48. What is Depth Testing?

A test that exercises a feature of a product in full detail is called Depth testing.

49. What is Dynamic Testing?

Testing software through executing it is called Dynamic testing. Also know about Static Testing.

50. What is Emulator?

202 Software Testing Interview Questions and Answers

Take your testing career to the next level, get ISTQB certified – http://istqbexamcertification.com

A device, computer program, or system that accepts the same inputs and produces the same outputs in a given system is called Emulator.

51. What is Endurance Testing?

Checks for memory leaks or other problems that may occur with prolonged execution.

52. What is End-to-End testing?

Testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate is called End-to-End testing.

53. What is Equivalence Class?

A portion of a component's input or output domains for which the component's behaviour is assumed to be the same from the component's specification.

54. What is Equivalence Partitioning?

A test case design technique for a component in which test cases are designed to execute representatives from equivalence classes.

55. What is Exhaustive Testing?

Testing which covers all combinations of input values and preconditions for an element of the software under test.

56. What is Functional Decomposition?

A technique used during planning, analysis and design; creates a functional hierarchy for the software.

57. What is Functional Specification?

A document that describes in detail the characteristics of the product with regard to its intended features is called Functional Specification.

58. What is Functional Testing?

Testing the features and operational behaviour of a product to ensure they correspond to its specifications. Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions. or Black Box Testing.

59. What is Glass Box Testing?

White Box Testing is also known as Glass Box testing.

60. What is Gorilla Testing?

Gorilla Testing is a testing technique in which sometimes developers also join hands with testers to test a particular module thoroughly in all aspects.

61. What is Gray Box Testing?

A combination of Black Box and White Box testing methodologies is called Gray box testing. Testing a piece of software against its specification but using some knowledge of its internal workings.

62. What is High Order Tests?

Black-box tests conducted once the software has been integrated.

63. What is Independent Test Group (ITG)?

202 Software Testing Interview Questions and Answers

Take your testing career to the next level, get ISTQB certified – http://istqbexamcertification.com

A group of people whose primary responsibility is software testing. Know more about Independent testing

64. What is Inspection?

A group review quality improvement process for written material. It consists of two aspects; product (document itself) improvement and process improvement (of both document production and inspection).

65. What is Integration Testing?

After integrating two different components together we do the integration testing. Integration testing is usually performed after unit and functional testing. This type of testing is especially relevant to client/server and distributed systems.

66. What is Installation Testing?

Installation testing is a kind of quality assurance work in the software industry that focuses on what customers will need to do to install and set up the new software successfully. The testing process may involve full, partial or upgrades install/uninstall processes.

67. What is Load Testing?

A load testing is a type of software testing which is conducted to understand the behaviour of the application under a specific expected load.

Also see Performance Testing.

68. What is Localization Testing?

This term refers to making software specifically designed for a specific locality.

69. What is Loop Testing?

A white box testing technique that exercises program loops in order to validate them.

70. What is Metric?

Metric is a standard of measurement. Software metrics are the statistics describing the structure or content of a program. A metric should be a real objective measurement of something such as number of bugs per lines of code.

71. What is Monkey Testing?

Testing a system or an Application on the fly, i.e just few tests here and there to ensure the system or an application does not crash out.

72. What is Negative Testing?

Testing aimed at showing software does not work. This is also known as "test to fail". See also Positive Testing.

73. What is Path Testing?

Testing in which all paths are in the program source code are tested at least once.

74. What is Performance Testing?

Testing conducted to evaluate the compliance of a system or component with specified performance requirements. Often this is performed using an automated test tool to simulate large number of users.

75. What is Positive Testing?

Testing aimed at showing software works. This is also known as "test to pass".

202 Software Testing Interview Questions and Answers

Take your testing career to the next level, get ISTQB certified – http://istqbexamcertification.com

76. What is Quality Assurance?

All those planned or systematic actions necessary to provide adequate confidence that a product or service is of the type and quality needed and expected by the customer.

77. What is Quality Audit?

A systematic and independent examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives.

78. What is Quality Circle?

A group of individuals with related interests that meet at regular intervals to consider problems or other matters related to the quality of outputs of a process and to the correction of problems or to the improvement of quality.

79. What is Quality Control?

The operational techniques and the activities used to fulfil and verify requirements of quality.

80. What is Quality Management?

That aspect of the overall management function that determines and implements the quality policy.

81. What is Quality Policy?

In quality management system, a quality policy is a document jointly developed by management and quality experts to express the quality objectives of the organization, the acceptable level of quality and the duties of specific departments to ensure quality.

82. What is Quality System?

The organizational structure, responsibilities, procedures, processes, and resources for implementing quality management is called Quality System.

83. What is Race Condition?

A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly.

84. What is Ramp Testing?

Continuously raising an input signal until the system breaks down.

85. What is Recovery Testing?

Recovery testing confirms that the program recovers from expected or unexpected events without loss of data or functionality. Events can include shortage of disk space, unexpected loss of communication, or power out conditions.

86. What is Regression Testing?

Retesting a previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made is called Regression testing.

87. What is Release Candidate?

A pre-release version, which contains the desired functionality of the final version, but which needs to be tested for bugs (which ideally should be removed before the final version is released).

88. What is Sanity Testing?

202 Software Testing Interview Questions and Answers

Take your testing career to the next level, get ISTQB certified – http://istqbexamcertification.com

Brief test of major functional elements of a piece of software to determine if its basically operational. See also Smoke Testing.

89. What is Scalability Testing?

Performance testing focused on ensuring the application under test gracefully handles increases in work load.

90. What is Security Testing?

Testing which confirms that the program can restrict access to authorized personnel and that the authorized personnel can access the functions available to their security level.

91. What is Smoke Testing?

In Software testing context, smoke testing refers to testing the basic functionality of the build.

92. What is Soak Testing?

Running a system at high load for a prolonged period of time is called Soak testing. For example; run several times more transactions in an entire day (or night) than would be expected in a busy day, to identify the performance problems that appear after a large number of transactions have been executed.

93. What is Software Requirements Specification?

A deliverable that describes all data, functional and behavioural requirements, all constraints, and all validation requirements for software/

94. What is Software Testing?

A set of activities conducted with the intent of finding errors in software.

95. What is Static Analysis?

Analysis of a program carried out without executing the program.

96. What is Static Analyzer?

A tool that carries out static analysis is called Static Analyzer.

97. What is Static Testing?

Analysis of a program carried out without executing the program.

98. What is Storage Testing?

Testing that verifies the program under test stores data files in the correct directories and that it reserves sufficient space to prevent unexpected termination resulting from lack of space. This is external storage as opposed to internal storage.

99. What is Stress Testing?

Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. Often this is performance testing using a very high level of simulated load.

100. What is Structural Testing?

Testing based on an analysis of internal workings and structure of a piece of software. See also White Box Testing.

101. What is System Testing?

Testing that attempts to discover defects that are properties of the entire system rather than of its individual components is called System testing.

102. What is Testability?

The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

103. What is Testing?

The process of exercising software to verify that it satisfies specified requirements and to detect errors is called testing. The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs), and to evaluate the features of the software item (Ref. IEEE Std 829). The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.

104. What is the difference between quality assurance and testing?

Quality assurance involves the entire software development process and testing involves operation of a system or application to evaluate the results under certain conditions. QA is oriented to prevention and Testing is oriented to detection.

105. Why does software have bugs?

Software have bugs because of the following reasons: 1) Miscommunication 2) programming errors 3) time pressures. 4) Changing requirements 5) software complexity

106. How do you do usability testing, security testing, installation testing, ADHOC, safety and smoke testing?

Usability testing : testing the user friendliness , simplicity of software. Security testing is testing the access rights for example admin has all rights, user1 has only right to read and open not to write . Installation testing is testing the software while installing it in different environment. Safety and security goes hand in hand most of the time it used logins. Smoke testing is to test the software whether the basic functionality is working or not. If u wants to Test the basic functionalities then we go for ADHOC it happens if we don’t have time to test procedure wise

107. What are memory leaks and buffer overflows?

Memory leaks means incomplete de-allocation - are bugs that happen very often. Buffer overflow means data sent as input to the server that overflows the boundaries of the input area, thus causing the server to misbehave.

108. What are the major differences between stress testing, load testing, Volume testing?

Stress testing means increasing the load and checking the performance at each level. Load testing means at a time giving more load by the expectation and checking the performance at that level. Volume testing means first we have to apply initial.

109. What is Exhaustive Testing?

Testing which covers all combinations of input values and preconditions for an element of the software under test.

110. What is Functional Decomposition?

A technique used during planning, analysis and design; creates a functional hierarchy for the software.

111. What is Functional Specification?

A document that describes in detail the characteristics of the product with regard to its intended features is called Functional Specification.

112. What is Test Bed?

An execution environment configured for testing. It may consist of specific hardware, OS, network topology, configuration of the product under test, other application or system software, etc. The Test Plan for a project should enumerate the test beds(s) to be used.

113. What is Test Case?

Test Case is a commonly used term for a specific test. This is usually the smallest unit of testing. A Test Case will consist of information such as requirements testing, test steps, verification steps, prerequisites, outputs, test environment, etc. A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test Driven Development? Testing methodology associated with Agile Programming in which every chunk of code is covered by unit tests, which must all pass all the time, in an effort to eliminate unit-level and regression bugs during development. Practitioners of TDD write a lot of tests, i.e. an equal number of lines of test code to the size of the production code.

114. What is Test Driver?

A program or test tool used to execute a test. Also known as a Test Harness.

115. What is Test Environment?

The hardware and software environment in which tests will be run, and any other software with which the software under test interacts when under test including stubs and test drivers.

116. What is Test First Design?

Test-first design is one of the mandatory practices of Extreme Programming (XP).It requires that programmers do not write any production code until they have first written a unit test.

117. What is Test Harness?

A program or test tool used to execute tests. Also known as a Test Driver.

118. What is Test Plan?

A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

119. What is Test Procedure?

A document providing detailed instructions for the execution of one or more test cases is called test Procedure.

120. What is Test Script?

Test script is commonly used to refer to the instructions for a particular test that will be carried out by an automated test tool.

121. What is Test Specification?

Test specification is a document specifying the test approach for a software feature or combination or features and the inputs, predicted results and execution conditions for the associated tests.

122. What is Test Suite?

A collection of tests used to validate the behaviour of a product. The scope of a Test Suite varies from organization to organization. There may be several Test Suites for a particular product for example. In most cases however a Test Suite is a high level concept, grouping together hundreds or thousands of tests related by what they are intended to test.

123. What is Test Tool?

Computer programs used for the testing of a system, a component of the system, or its documentation is called a test tool.

124. What is Thread Testing?

A variation of top-down testing where the progressive integration of components follows the implementation of subsets of the requirements, as opposed to the integration of components by successively lower levels.

125. What is Top Down Testing?

Top down testing is an approach to integration testing where the component at the top of the component hierarchy is tested first, with lower level components being simulated by stubs. Tested components are then used to test lower level components. The process is repeated until the lowest level components have been tested.

126. What is Total Quality Management?

Total Quality Management is a company commitment to develop a process that achieves high quality product and customer satisfaction.

127. What is Traceability Matrix?

Traceability Matrix is a document showing the relationship between Test Requirements and Test Cases.

128. What is Usability Testing?

Testing the ease with which users can learn and use a product.

129. What is Use Case?

Use case is the specification of tests that are conducted from the end-user perspective. Use cases tend to focus on operating software as an end-user would conduct their day-to-day activities.

130. What is Unit Testing?

Testing of individual software components is called Unit testing.

131. What is Validation?

The process of evaluating software at the end of the software development process to ensure compliance with software requirements is called Validation.

132. What is Verification?

The process of determining whether or not the products of a given phase of the software development cycle meet the implementation steps and can be traced to the incoming objectives established during the previous phase.

133. What is White Box Testing?

Testing based on an analysis of internal workings and structure of a piece of software. This includes techniques such as Branch Testing and Path Testing. This also known as Structural Testing and Glass Box Testing. Contrast with Black Box Testing. White box testing is used to test the internal logic

of the code for ex checking whether the path has been executed once, checking whether the branches has been executed at least once .This is used to check the structure of the code.

134. What is Workflow Testing?

Workflow processes technique in software testing by routing a record through each possible path. These tests are performed to ensure that each workflow process accurately reflects the business process. This kind of testing holds good for workflow-based applications.

135. What's the difference between load and stress testing ?

One of the most common, but unfortunate misuse of terminology is treating “load testing” and “stress testing” as synonymous. The consequence of this ignorant semantic abuse is usually that the system is neither properly “load tested” nor subjected to a meaningful stress test. Stress testing is subjecting a system to an unreasonable load while denying it the resources (e.g., RAM, disc, mips, interrupts, etc.) needed to process that load. The idea is to stress a system to the breaking point in order to find bugs that will make that break potentially harmful. The system is not expected to process the overload without adequate resources, but to behave (e.g., fail) in a decent manner (e.g., not corrupting or losing data). Bugs and failure modes discovered under stress testing may or may not be repaired depending on the application, the failure mode, consequences, etc. The load (incoming transaction stream) in stress testing is often deliberately distorted so as to force the system into resource depletion. Load testing is subjecting a system to a statistically representative (usually) load. The two main reasons for using such loads is in support of software reliability testing and in performance testing. The term 'load testing' by itself is too vague and imprecise to warrant use. For example, do you mean representative load,' 'overload,' 'high load,' etc. In performance testing, load is varied from a minimum (zero) to the maximum level the system can sustain without running out of resources or having, transactions >suffer (application-specific) excessive delay. A third use of the term is as a test whose objective is to determine the maximum sustainable load the system can handle. In this usage, 'load testing' is merely testing at the highest transaction arrival rate in performance testing.

136. What's the difference between QA and testing?

QA is more a preventive thing, ensuring quality in the company and therefore the product rather than just testing the product for software bugs? TESTING means 'quality control' QUALITY CONTROL measures the quality of a product QUALITY ASSURANCE measures the quality of processes used to create a quality product.

137. What is the best tester to developer ratio?

Tester : developer ratios range from 10:1 to 1:10. There's no simple answer. It depends on so many things, Amount of reused code, number and type of interfaces, platform, quality goals, etc. It also can depend on the development model. The more specs, the less testers. The roles can play a big part also. Does QA own beta? Do you include process auditors or planning activities? These figures can all vary very widely depending on how you define 'tester' and 'developer'. In some organizations, a 'tester' is anyone who happens to be testing software at the time -- such as their own. In other organizations, a 'tester' is only a member of an independent test group. It is better to ask about the test labour content than it is to ask about the tester/developer ratio. The test labour content, across most applications is generally accepted as 50%, when people do honest accounting. For life-critical software, this can go up to 80%.

138. How can new Software QA processes be introduced in an existing organization?

- A lot depends on the size of the organization and the risks involved. For large organizations with high-risk (in terms of lives or property) projects, serious management buy-in is required and a formalized QA process is necessary. - Where the risk is lower, management and organizational buy-in and QA implementation may be a slower, step-at-a-time process. QA processes should be balanced with productivity so as to keep bureaucracy from getting out of hand. - For small groups or projects, a

more ad-hoc process may be appropriate, depending on the type of customers and projects. A lot will depend on team leads or managers, feedback to developers, and ensuring adequate communications among customers, managers, developers, and testers. - In all cases the most value for effort will be in requirements management processes, with a goal of clear, complete, testable requirement specifications or expectations.

139. What are 5 common problems in the software development process?

The 5 common problems in the software development process are as follows:

1) Poor requirements - if requirements are unclear, incomplete, too general, or not testable, there will be problems. 2) Unrealistic schedule - if too much work is crammed in too little time, problems are inevitable. 3) Inadequate testing - no one will know whether or not the program is any good until the customer complaints or systems crash. 4) Features - requests to pile on new features after development is underway; extremely common. 5) Miscommunication - if developers don't know what's needed or customer's have erroneous expectations, problems are guaranteed.

140. What are 5 common solutions to software development problems?

The 5 common solutions to software development problems as given below:

1) Solid requirements - clear, complete, detailed, cohesive, attainable, testable requirements that are agreed to by all players. Use prototypes to help nail down requirements. 2) Realistic schedules - allow adequate time for planning, design, testing, bug fixing, re-testing, changes, and documentation; personnel should be able to complete the project without burning out. 3) Adequate testing - start testing early on, re-test after fixes or changes, plan for adequate time for testing and bug-fixing. 4) Stick to initial requirements as much as possible - be prepared to defend against changes and additions once development has begun, and be prepared to explain consequences. If changes are necessary, they should be adequately reflected in related schedule changes. If possible, use rapid prototyping during the design phase so that customers can see what to expect. This will provide them a higher comfort level with their requirements decisions and minimize changes later on. 5) communication - require walkthroughs and inspections when appropriate; make extensive use of group communication tools - e-mail, groupware, networked bug-tracking tools and change management tools, intranet capabilities, etc.; insure that documentation is available and up-to-date - preferably electronic, not paper; promote teamwork and cooperation; use prototypes early on so that customers' expectations are clarified.

141. What is a 'good code'?

'Good code' is a code that works, is bug free, and is readable and maintainable. Some organizations have coding 'standards' that all developers are supposed to adhere to, but everyone has different ideas about what's best, or what is too many or too few rules. There are also various theories and metrics, such as McCabe Complexity metrics. It should be kept in mind that excessive use of standards and rules can stifle productivity and creativity. 'Peer reviews', 'buddy checks' code analysis tools, etc. can be used to check for problems and enforce standards. For C and C++ coding, here are some typical ideas to consider in setting rules/standards; these may or may not apply to a particular situation: - minimize or eliminate use of global variables. - use descriptive function and method names - use both upper and lower case, avoid abbreviations, use as many characters as necessary to be adequately descriptive (use of more than 20 characters is not out of line); be consistent in naming conventions. - use descriptive variable names - use both upper and lower case, avoid abbreviations, use as many characters as necessary to be adequately descriptive (use of more than 20 characters is not out of line); be consistent in naming conventions. - function and method sizes should be minimized; less than 100 lines of code is good, less than 50 lines is preferable. - function descriptions should be clearly spelled out in comments preceding a function's code.- organize code for readability. - use whitespace generously - vertically and horizontally - each line of code should contain 70 characters max. - one code statement per line. - coding style should be consistent throughout a program (eg, use of brackets, indentations, naming conventions, etc.) - in adding comments, err on

the side of too many rather than too few comments; a common rule of thumb is that there should be at least as many lines of comments (including header blocks) as lines of code. - no matter how small, an application should include documentation of the overall program function and flow (even a few paragraphs is better than nothing); or if possible a separate flow chart and detailed program documentation. - make extensive use of error handling procedures and status and error logging. - For C++, to minimize complexity and increase maintainability, avoid too many levels of inheritance in class hierarchies (relative to the size and complexity of the application). Minimize use of multiple inheritance, and minimize use of operator overloading (note that the Java programming language eliminates multiple inheritance and operator overloading.) - For C++, keep class methods small, less than 50 lines of code per method is preferable. - For C++, make liberal use of exception handlers

142. What is a 'good design'?

'Design' could refer to many things, but often refers to 'functional design' or 'internal design'. Good internal design is indicated by software code whose overall structure is clear, understandable, easily modifiable, and maintainable; is robust with sufficient error-handling and status logging capability; and works correctly when implemented. Good functional design is indicated by an application whose functionality can be traced back to customer and end-user requirements. For programs that have a user interface, it's often a good idea to assume that the end user will have little computer knowledge and may not read a user manual or even the on-line help; some common rules-of-thumb include: - the program should act in a way that least surprises the user - it should always be evident to the user what can be done next and how to exit - the program shouldn't let the users do something stupid without warning them. Know more about test design, test design technique, categories of test design technique and test design tools

143. What makes a good test engineer?

A good test engineer has a 'test to break' attitude, an ability to take the point of view of the customer, a strong desire for quality, and an attention to detail. Tact and diplomacy are useful in maintaining a cooperative relationship with developers, and an ability to communicate with both technical (developers) and non-technical (customers, management) people is useful. Previous software development experience can be helpful as it provides a deeper understanding of the software development process, gives the tester an appreciation for the developers' point of view, and reduce the learning curve in automated test tool programming. Judgment skills are needed to assess high-risk areas of an application on which to focus testing efforts when time is limited.

144. What makes a good Software QA engineer?

The same qualities a good tester has are useful for a QA engineer. Additionally, they must be able to understand the entire software development process and how it can fit into the business approach and goals of the organization. Communication skills and the ability to understand various sides of issues are important. In organizations in the early stages of implementing QA processes, patience and diplomacy are especially needed. An ability to find problems as well as to see 'what's missing' is important for inspections and reviews.

145. What makes a good QA or Test manager?

A good QA, test, or QA/Test(combined) manager should: - be familiar with the software development process - be able to maintain enthusiasm of their team and promote a positive atmosphere, despite what is a somewhat 'negative' process (e.g., looking for or preventing problems) - be able to promote teamwork to increase productivity - be able to promote cooperation between software, test, and QA engineers - have the diplomatic skills needed to promote improvements in QA processes -have the ability to withstand pressures and say 'no' to other managers when quality is insufficient or QA processes are not being adhered to - have people judgement skills for hiring and keeping skilled

personnel- be able to communicate with technical and non-technical people, engineers, managers, and customers. - be able to run meetings and keep them focused

146. What's the role of documentation in QA?

The role of documentation in QA is Critical. (Note that documentation can be electronic, not necessarily paper.) QA practices should be documented such that they are repeatable. Specifications, designs, business rules, inspection reports, configurations, code changes, test plans, test cases, bug reports, user manuals, etc. should all be documented. There should ideally be a system for easily finding and obtaining documents and determining what documentation will have a particular piece of information. Change management for documentation should be used if possible. Know more about documentation testing

147. What's the big deal about 'requirements'?

One of the most reliable methods of insuring problems, or failure, in a complex software project is to have poorly documented requirements specifications. Requirements are the details describing an application's externally-perceived functionality and properties. Requirements should be clear, complete, reasonably detailed, cohesive, attainable, and testable. A non-testable requirement would be, for example, 'user-friendly' (too subjective). A testable requirement would be something like 'the user must enter their previously-assigned password to access the application'. Determining and organizing requirements details in a useful and efficient way can be a difficult effort; different methods are available depending on the particular project. Many books are available that describe various approaches to this task. Care should be taken to involve ALL of a project's significant 'customers' in the requirements process. 'Customers' could be in-house personnel or out, and could include end-users, customer acceptance testers, customer contract officers, customer management, future software maintenance engineers, salespeople, etc. Anyone who could later derail the project if their expectations aren't met should be included if possible. Organizations vary considerably in their handling of requirements specifications. Ideally, the requirements are spelled out in a document with statements such as 'The product shall.....'. 'Design' specifications should not be confused with 'requirements'; design specifications should be traceable back to the requirements. In some organizations requirements may end up in high level project plans, functional specification documents, in design documents, or in other documents at various levels of detail. No matter what they are called, some type of documentation with detailed requirements will be needed by testers in order to properly plan and execute tests. Without such documentation, there will be no clear-cut way to determine if a software application is performing correctly.

148. What steps are needed to develop and run software tests?

The following are some of the steps to consider: - Obtain requirements, functional design, and internal design specifications and other necessary documents - Obtain budget and schedule requirements - Determine project-related personnel and their responsibilities, reporting requirements, required standards and processes (such as release processes, change processes, etc.) - Identify application's higher-risk aspects, set priorities, and determine scope and limitations of tests - Determine test approaches and methods - unit, integration, functional, system, load, usability tests, etc. - Determine test environment requirements (hardware, software, communications, etc.) -Determine testware requirements (record/playback tools, coverage analyzers, test tracking, problem/bug tracking, etc.) - Determine test input data requirements - Identify tasks, those responsible for tasks, and labor requirements - Set schedule estimates, timelines, milestones - Determine input equivalence classes, boundary value analyses, error classes - Prepare test plan document and have needed reviews/approvals - Write test cases - Have needed reviews/inspections/approvals of test cases - Prepare test environment and testware, obtain needed user manuals/reference documents/configuration guides/installation guides, set up test tracking processes, set up logging and archiving processes, set up or obtain test input data - Obtain and install software releases - Perform

tests - Evaluate and report results - Track problems/bugs and fixes - Retest as needed - Maintain and update test plans, test cases, test environment, and testware through life cycle

149. What is 'configuration management'?

Configuration management covers the processes used to control, coordinate, and track: code, requirements, documentation, problems, change requests, designs, tools/compilers/libraries/patches, changes made to them, and who makes the changes.

150. What if the software is so buggy it can't really be tested at all?

The best bet in this situation is for the testers to go through the process of reporting whatever bugs or blocking-type problems initially show up, with the focus being on critical bugs. Since this type of problem can severely affect schedules, and indicates deeper problems in the software development process (such as insufficient unit testing or insufficient integration testing, poor design, improper build or release procedures, etc.) managers should be notified, and provided with some documentation as evidence of the problem. Know more about Severity and Priority

151. How can it be known when to stop testing?

This can be difficult to determine. Many modern software applications are so complex, and run in such an interdependent environment, that complete testing can never be done. Common factors in deciding when to stop are: - Deadlines (release deadlines, testing deadlines, etc.)- Test cases completed with certain percentage passed - Test budget depleted - Coverage of code/functionality/requirements reaches a specified point - Bug rate falls below a certain level - Beta or alpha testing period ends

152. What if there isn't enough time for thorough testing?

Use risk analysis to determine where testing should be focused. Since it's rarely possible to test every possible aspect of an application, every possible combination of events, every dependency, or everything that could go wrong, risk analysis is appropriate to most software development projects. This requires judgement skills, common sense, and experience. (If warranted, formal methods are also available.) Considerations can include: - Which functionality is most important to the project's intended purpose? - Which functionality is most visible to the user? - Which functionality has the largest safety impact? - Which functionality has the largest financial impact on users? - Which aspects of the application are most important to the customer? - Which aspects of the application can be tested early in the development cycle? - Which parts of the code are most complex, and thus most subject to errors? - Which parts of the application were developed in rush or panic mode? - Which aspects of similar/related previous projects caused problems? - Which aspects of similar/related previous projects had large maintenance expenses? - Which parts of the requirements and design are unclear or poorly thought out? - What do the developers think are the highest-risk aspects of the application? - What kinds of problems would cause the worst publicity? - What kinds of problems would cause the most customer service complaints?- What kinds of tests could easily cover multiple functionalities? - Which tests will have the best high-risk-coverage to time-required ratio?

153. What can be done if requirements are changing continuously?

This is a common problem and a major headache. - Work with the project's stakeholders early on to understand how requirements might change so that alternate test plans and strategies can be worked out in advance, if possible. - It's helpful if the application's initial design allows for some adaptability so that later changes do not require redoing the application from scratch. - If the code is well-commented and well-documented this makes changes easier for the developers.- Use rapid prototyping whenever possible to help customers feel sure of their requirements and minimize changes. - The project's initial schedule should allow for some extra time commensurate with the possibility of changes.- Try to move new requirements to a 'Phase 2' version of an application, while using the original requirements for the 'Phase 1' version. - Negotiate to allow only easily-implemented new requirements into the

project, while moving more difficult new requirements into future versions of the application. - Be sure that customers and management understand the scheduling impacts, inherent risks, and costs of significant requirements changes. Then let management or the customers (not the developers or testers) decide if the changes are warranted - after all, that's their job. - Balance the effort put into setting up automated testing with the expected effort required to re-do them to deal with changes. - Try to design some flexibility into automated test scripts. - Focus initial automated testing on application aspects that are most likely to remain unchanged. - Devote appropriate effort to risk analysis of changes to minimize regression testing needs. - Design some flexibility into test cases (this is not easily done; the best bet might be to minimize the detail in the test cases, or set up only higher-level generic-type test plans) - Focus less on detailed test plans and test cases and more on ad hoc testing (with an understanding of the added risk that this entails).

154. What if the project isn't big enough to justify extensive testing?

Consider the impact of project errors, not the size of the project. However, if extensive testing is still not justified, risk analysis is again needed and the same considerations as described previously in 'What if there isn't enough time for thorough testing?' apply. The tester might then do ad hoc testing, or write up a limited test plan based on the risk analysis.

155. What if the application has functionality that wasn't in the requirements?

It may take serious effort to determine if an application has significant unexpected or hidden functionality, and it would indicate deeper problems in the software development process. If the functionality isn't necessary to the purpose of the application, it should be removed, as it may have unknown impacts or dependencies that were not taken into account by the designer or the customer. If not removed, design information will be needed to determine added testing needs or regression testing needs. Management should be made aware of any significant added risks as a result of the unexpected functionality. If the functionality only effects areas such as minor improvements in the user interface, for example, it may not be a significant risk.

156. How can Software QA processes be implemented without stifling productivity?

By implementing QA processes slowly over time, using consensus to reach agreement on processes, and adjusting and experimenting as an organization grows and matures, productivity will be improved instead of stifled. Problem prevention will lessen the need for problem detection, panics and burn-out will decrease, and there will be improved focus and less wasted effort. At the same time, attempts should be made to keep processes simple and efficient, minimize paperwork, promote computer-based processes and automated tracking and reporting, minimize time required in meetings, and promote training as part of the QA process. However, no one - especially talented technical types - likes rules or bureaucracy, and in the short run things may slow down a bit. A typical scenario would be that more days of planning and development will be needed, but less time will be required for late-night bug-fixing and calming of irate customers.

157. What if an organization is growing so fast that fixed QA processes are impossible?

This is a common problem in the software industry, especially in new technology areas. There is no easy solution in this situation, other than: - Hire good people - Management should 'ruthlessly prioritize' quality issues and maintain focus on the customer - Everyone in the organization should be clear on what 'quality' means to the customer

158. How does a client/server environment affect testing?

Client/server applications can be quite complex due to the multiple dependencies among clients, data communications, hardware, and servers. Thus testing requirements can be extensive. When time is limited (as it usually is) the focus should be on integration and system testing. Additionally, load/stress/performance testing may be useful in determining client/server application limitations and capabilities. There are commercial tools to assist with such testing.

159. How can World Wide Web sites be tested?

Web sites are essentially client/server applications - with web servers and 'browser' clients. Consideration should be given to the interactions between html pages, TCP/IP communications, Internet connections, firewalls, applications that run in web pages (such as applets, javascript, plug-in applications), and applications that run on the server side (such as cgi scripts, database interfaces, logging applications, dynamic page generators, asp, etc.). Additionally, there are a wide variety of servers and browsers, various versions of each, small but sometimes significant differences between them, variations in connection speeds, rapidly changing technologies, and multiple standards and protocols. The end result is that testing for web sites can become a major ongoing effort. Other considerations might include: - What are the expected loads on the server (e.g., number of hits per unit time?), and what kind of performance is required under such loads (such as web server response time, database query response times). What kinds of tools will be needed for performance testing (such as web testing tools, other tools already in house that can be adapted, web robot downloading tools, etc.)? - Who is the target audience? What kind of browsers will they be using? What kind of connection speed will they by using? Are they intra- organization (thus with likely high connection speeds and similar browsers) or Internet-wide (thus with a wide variety of connection speeds and browser types)? - What kind of performance is expected on the client side (e.g., how fast should pages appear, how fast should animations, applets, etc. load and run)? - Will down time for server and content maintenance/upgrades be allowed? How much? - What kinds of security (firewalls, encryptions, passwords, etc.) will be required and what is it expected to do? How can it be tested? - How reliable are the site's Internet connections required to be? And how does that affect backup system or redundant connection requirements and testing? - What processes will be required to manage updates to the web site's content, and what are the requirements for maintaining, tracking, and controlling page content, graphics, links, etc.? - Which HTML specification will be adhered to? How strictly? What variations will be allowed for targeted browsers? - Will there be any standards or requirements for page appearance and/or graphics throughout a site or parts of a site?? - How will internal and external links be validated and updated? how often? - Can testing be done on the production system, or will a separate test system be required? How are browser caching, variations in browser option settings, dial-up connection variability, and real-world internet 'traffic congestion' problems to be accounted for in testing?- How extensive or customized are the server logging and reporting requirements; are they considered an integral part of the system and do they require testing?- How are cgi programs, applets, java scripts, ActiveX components, etc. to be maintained, tracked, controlled, and tested? - Pages should be 3-5 screens max unless content is tightly focused on a single topic. If larger, provide internal links within the page. - The page layouts and design elements should be consistent throughout a site, so that it's clear to the user that they're still within a site. - Pages should be as browser-independent as possible, or pages should be provided or generated based on the browser-type. - All pages should have links external to the page; there should be no dead-end pages. - The page owner, revision date, and a link to a contact person or organization should be included on each page.

160. How is testing affected by object-oriented designs?

Well-engineered object-oriented design can make it easier to trace from code to internal design to functional design to requirements. While there will be little affect on black box testing (where an understanding of the internal design of the application is unnecessary), white-box testing can be oriented to the application's objects. If the application was well-designed this can simplify test design.

161. What is Extreme Programming and what's it got to do with testing?

Extreme Programming (XP) is a software development approach for small teams on risk-prone projects with unstable requirements. It was created by Kent Beck who described the approach in his book 'Extreme Programming Explained'. Testing ('extreme testing') is a core aspect of Extreme Programming. Programmers are expected to write unit and functional test code first - before the

application is developed. Test code is under source control along with the rest of the code. Customers are expected to be an integral part of the project team and to help develop scenarios for acceptance/black box testing. Acceptance tests are preferably automated, and are modified and rerun for each of the frequent development iterations. QA and test personnel are also required to be an integral part of the project team. Detailed requirements documentation is not used, and frequent re-scheduling, re-estimating, and re-prioritizing is expected.

162. Will automated testing tools make testing easier?

- Possibly. For small projects, the time needed to learn and implement them may not be worth it. For larger projects, or on-going long-term projects they can be valuable. - A common type of automated tool is the 'record/playback' type. For example, a tester could click through all combinations of menu choices, dialog box choices, buttons, etc. in an application GUI and have them 'recorded' and the results logged by a tool. The 'recording' is typically in the form of text based on a scripting language that is interpretable by the testing tool. If new buttons are added, or some underlying code in the application is changed, etc. the application can then be retested by just 'playing back' the 'recorded' actions, and comparing the logging results to check effects of the changes. The problem with such tools is that if there are continual changes to the system being tested, the 'recordings' may have to be changed so much that it becomes very time-consuming to continuously update the scripts. Additionally, interpretation of results (screens, data, logs, etc.) can be a difficult task. Note that there are record/playback tools for text-based interfaces also, and for all types of platforms.- Other automated tools can include: code analyzers - monitor code complexity, adherence to standards, etc. coverage analyzers - these tools check which parts of the code have been exercised by a test, and may be oriented to code statement coverage, condition coverage, path coverage, etc. memory analyzers - such as bounds-checkers and leak detectors. load/performance test tools - for testing client/server and web applications under various load levels. web test tools - to check that links are valid, HTML code usage is correct, client-side and server-side programs work, a web site's interactions are secure. other tools - for test case management, documentation management, bug reporting, and configuration management.

163. What's the difference between black box and white box testing?

Black-box and white-box are test design methods. Black-box test design treats the system as a “black-box”, so it doesn't explicitly use knowledge of the internal structure. Black-box test design is usually described as focusing on testing functional requirements. Synonyms for black-box include: behavioural, functional, opaque-box, and closed-box. White-box test design allows one to peek inside the “box”, and it focuses specifically on using internal knowledge of the software to guide the selection of test data. Synonyms for white-box include: structural, glass-box and clear-box. While black-box and white-box are terms that are still in popular use, many people prefer the terms 'behavioural' and 'structural'. Behavioural test design is slightly different from black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged. In practice, it hasn't proven useful to use a single test design method. One has to use a mixture of different methods so that they aren't hindered by the limitations of a particular one. Some call this 'gray-box' or 'translucent-box' test design, but others wish we'd stop talking about boxes altogether. It is important to understand that these methods are used during the test design phase, and their influence is hard to see in the tests once they're implemented. Note that any level of testing (unit testing, system testing, etc.) can use any test design methods. Unit testing is usually associated with structural test design, but this is because testers usually don't have well-defined requirements at the unit level to validate.

164. What kinds of testing should be considered?

Below are the kinds of testing which should be considered:

Black box testing can be considered which is not based on any knowledge of internal design or code. Tests are based on requirements and functionality. White box testing can also be considered which is based on knowledge of the internal logic of an application's code. Tests are based on coverage of

code statements, branches, paths, conditions. Unit testing - the most 'micro' scale of testing; to test particular functions or code modules. Unit testing is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses. incremental integration testing - continuous testing of an application as new functionality is added; requires that various aspects of an application's functionality be independent enough to work separately before all parts of the program are completed, or that test drivers be developed as needed; done by programmers or by testers. integration testing - testing of combined parts of an application to determine if they function together correctly. The 'parts' can be code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems. functional testing - black-box type testing geared to functional requirements of an application; this type of testing should be done by testers. This doesn't mean that the programmers shouldn't check that their code works before releasing it (which of course applies to any stage of testing.) system testing - black-box type testing that is based on overall requirements specifications; covers all combined parts of a system. end-to-end testing - similar to system testing; the 'macro' end of the test scale; involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate. sanity testing or smoke testing - typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort. For example, if the new software is crashing systems every 5 minutes, bogging down systems to a crawl, or corrupting databases, the software may not be in a 'sane' enough condition to warrant further testing in its current state. regression testing - re-testing after fixes or modifications of the software or its environment. It can be difficult to determine how much re-testing is needed, especially near the end of the development cycle. Automated testing tools can be especially useful for this type of testing. acceptance testing - final testing based on specifications of the end-user or customer, or based on use by end-users/customers over some limited period of time. load testing - testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails. stress testing - term often used interchangeably with 'load' and 'performance' testing. Also used to describe such tests as system functional testing while under unusually heavy loads, heavy repetition of certain actions or inputs, input of large numerical values, large complex queries to a database system, etc. performance testing - term often used interchangeably with 'stress' and 'load' testing. Ideally 'performance' testing (and any other 'type' of testing) is defined in requirements documentation or QA or Test Plans. Usability testing - testing for 'user-friendliness'. Clearly this is subjective, and will depend on the targeted end-user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers. install/uninstall testing - testing of full, partial, or upgrade install/uninstall processes. recovery testing - testing how well a system recovers from crashes, hardware failures, or other catastrophic problems. failover testing - typically used interchangeably with 'recovery testing' security testing - testing how well the system protects against unauthorized internal or external access, wilful damage, etc; may require sophisticated testing techniques. Compatibility testing - testing how well software performs in a particular hardware/software/operating system/network/etc. environment. Exploratory testing - often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it. Ad-hoc testing - similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it. Context-driven testing - testing driven by an understanding of the environment, culture, and intended use of software. For example, the testing approach for life-critical medical equipment software would be completely different than that for a low-cost computer game. User acceptance testing is done to determine whether the software is satisfactory to an end-user or customer. Comparison testing is about comparing software weaknesses and strengths to competing products. Alpha testing - testing of an application when development is nearing completion; minor design changes may still be made as

a result of such testing. Alpha testing is typically done by end-users or others, not by programmers or testers. Beta testing - testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Beta testing is typically done by end-users or others, not by programmers or testers. mutation testing - a method for determining if a set of test data or test cases is useful, by deliberately introducing various code changes ('bugs') and retesting with the original test data/cases to determine if the 'bugs' are detected. Proper implementation requires large computational resources.

165. Why is it often hard for management to get serious about quality assurance?

Solving problems is a high-visibility process; preventing problems is low-visibility. This is illustrated by an old parable: In ancient China there was a family of healers, one of whom was known throughout the land and employed as a physician to a great lord. The physician was asked which of his family was the most skillful healer. He replied, "I tend to the sick and dying with drastic and dramatic treatments, and on occasion someone is cured and my name gets out among the lords.""My elder brother cures sickness when it just begins to take root, and his skills are known among the local peasants and neighbours." "My eldest brother is able to sense the spirit of sickness and eradicate it before it takes form. His name is unknown outside our home."

166. Why does software have bugs?

1) Miscommunication or no communication - as to specifics of what an application should or shouldn't do (the application's requirements). 2) Software complexity - the complexity of current software applications can be difficult to comprehend for anyone without experience in modern-day software development. Multi-tiered applications, client-server and distributed applications, data communications, enormous relational databases, and sheer size of applications have all contributed to the exponential growth in software/system complexity. programming errors - programmers, like anyone else, can make mistakes. 3) Changing requirements (whether documented or undocumented) - the end-user may not understand the effects of changes, or may understand and request them anyway - redesign, rescheduling of engineers, effects on other projects, work already completed that may have to be redone or thrown out, hardware requirements that may be affected, etc. If there are many minor changes or any major changes, known and unknown dependencies among parts of the project are likely to interact and cause problems, and the complexity of coordinating changes may result in errors. Enthusiasm of engineering staff may be affected. In some fast-changing business environments, continuously modified requirements may be a fact of life. In this case, management must understand the resulting risks, and QA and test engineers must adapt and plan for continuous extensive testing to keep the inevitable bugs from running out of control. 4) Poorly documented code - it's tough to maintain and modify code that is badly written or poorly documented; the result is bugs. In many organizations management provides no incentive for programmers to document their code or write clear, understandable, maintainable code. In fact, it's usually the opposite: they get points mostly for quickly turning out code, and there's job security if nobody else can understand it ('if it was hard to write, it should be hard to read'). 5) software development tools - visual tools, class libraries, compilers, scripting tools, etc. often introduce their own bugs or are poorly documented, resulting in added bugs. Also know about from where do defects and failures in software testing arise?

167. How can new Software QA processes be introduced in an existing organization?

A lot depends on the size of the organization and the risks involved. For large organizations with high-risk (in terms of lives or property) projects, serious management buy-in is required and a formalized QA process is necessary. Where the risk is lower, management and organizational buy-in and QA implementation may be a slower, step-at-a-time process. QA processes should be balanced with productivity so as to keep bureaucracy from getting out of hand. For small groups or projects, a more ad-hoc process may be appropriate, depending on the type of customers and projects. A lot will depend on team leads or managers, feedback to developers, and ensuring adequate communications among customers, managers, developers, and testers. The most value for effort will often be in (a)

requirements management processes, with a goal of clear, complete, testable requirement specifications embodied in requirements or design documentation, or in 'agile'-type environments extensive continuous coordination with end-users, (b) design inspections and code inspections, and (c) post-mortems/retrospectives.

168. How do the companies expect the defect reporting to be communicated by the tester to the development team. Can the excel sheet template be used for defect reporting. If so what are the common fields that are included? Who assigns the priority and severity of the defect?

To report bugs in excel: S.no, Module Screen/ Section Issue detail, Severity, Priority, Issue status this is how to report bugs in excel sheet and also set filters on the Columns attributes. But most of the companies use the share point process of reporting bugs In this when the project came for testing a module wise detail of project is inserted to the defect management system they are using. It contains following field1) Date 2) Issue brief 3) Issue description (used for developer to regenerate the issue) 4) Issue status (active, resolved, on hold, suspend and not able to regenerate) 5) Assign to (Names of members allocated to project) 6) Priority (High, medium and low) 7) severity (Major, medium and low)

169. What are the tables in test plans and test cases?

Test plan is a document that contains the scope, approach, test design and test strategies. It includes the following:-1) Test case identifier 2) Scope 3) Features to be tested 4) Features not to be tested. 5) Test strategy. 6) Test Approach 7) Test Deliverables 8) Responsibilities. 9) Staffing and Training 10) Risk and Contingencies 11) Approval While A test case is a noted/documented set of steps/activities that are carried out or executed on the software in order to confirm its functionality/behaviour to certain set of inputs.

170. What are the table contents in test plans and test cases?

Test Plan is a document which is prepared with the details of the testing priority. A test Plan generally includes: 1) Objective of Testing 2) Scope of Testing 3) Reason for testing 4) Timeframe 5) Environment 6) Entry and exit criteria 7) Risk factors involved 8) Deliverables

171. What automating testing tools are you familiar with?

Win Runner, Load runner, QTP , Silk Performer, Test director, Rational robot, QA run.

172. Why did you use automating testing tools in your job?

Automating testing tools are used because of the following reasons:

1) For regression testing 2) Criteria to decide the condition of a particular build

173. How do you plan test automation?

1) Prepare the automation Test plan 2) Identify the scenario 3) Record the scenario 4) Enhance the scripts by inserting check points and Conditional Loops 5) Incorporated Error Handler 6) Debug the script 7) Fix the issue 8) Rerun the script and report the result.

174. Can test automation improve test effectiveness?

Yes, automating a test makes the test process: 1) Fast 2) Reliable 3) Repeatable 4) Programmable 5) Reusable 6) Comprehensive

175. What are the main attributes of test automation?

software test automation attributes : Maintainability - the effort needed to update the test automation suites for each new release Reliability - the accuracy and repeatability of the test automation Flexibility - the ease of working with all the different kinds of automation test ware Efficiency - the total cost related to the effort needed for the automation Portability - the ability of the automated test to run on different environments Robustness - the effectiveness of automation on an unstable or rapidly changing system Usability - the extent to which automation can be used by different types of users

202 Software Testing Interview Questions and Answers

Take your testing career to the next level, get ISTQB certified – http://istqbexamcertification.com

176. Does automation replace manual testing?

There can be some functionality which cannot be tested in an automated tool so we may have to do it manually. Therefore manual testing can never be replaced. (We can write the scripts for negative testing also but it is hectic task).When we talk about real environment we do negative testing manually.

177. How will you choose a tool for test automation?

Choosing of a tool depends on many things like 1) Application to be tested 2) Test environment 3) Scope and limitation of the tool. 4) Feature of the tool 5) Cost of the tool 6) Whether the tool is compatible with your application which means tool should be able to interact with your application 7) Ease of use

178. How you will evaluate the tool for test automation?

We need to concentrate on the features of the tools and how this could be beneficial for our project. The additional new features and the enhancements of the features will also help.

179. What are main benefits of test automation?

FAST, RELIABLE, COMPREHENSIVE, REUSABLE

180. What could go wrong with test automation?

1. The choice of automation tool for certain technologies 2. Wrong set of test automated

181. How you will describe testing activities?

Testing activities start from the elaboration phase. The various testing activities are preparing the test plan, Preparing test cases, Execute the test case, Log the bug, validate the bug & take appropriate action for the bug, Automate the test cases.

182. What testing activities you may want to automate?

Automate all the high priority test cases which need to be executed as a part of regression testing for each build cycle.

183. Describe common problems of test automation.

The common problems are: 1. Maintenance of the old script when there is a feature change or enhancement 2. The change in technology of the application will affect the old scripts

184. What types of scripting techniques for test automation do you know?

5 types of scripting techniques: Linear Structured Shared Data Driven Key Driven

185. What are principles of good testing scripts for automation?

1) Proper code guiding standards 2) Standard format for defining functions, exception handler etc 3) Comments for functions 4.)Proper error handling mechanisms 5) The appropriate synchronisation techniques.

186. Can the activities of test case design be automated?

As I know it, test case design is about formulating the steps to be carried out to verify something about the application under test. And this cannot be automated. However, I agree that the process of putting the test results into the excel sheet.

187. What are the limitations of automating software testing?

Hard-to-create environments like “out of memory”, “invalid input/reply”, and “corrupt registry entries” make applications behave poorly and existing automated tools can’t force these condition - they simply test your application in “normal” environment.

188. What skills needed to be a good test automator?

1) Good Logic for programming. 2) Strong analytical skills 3) Pessimistic in Nature.

189. How to find that tools work well with your existing system?

1. Discuss with the support officials 2. Download the trial version of the tool and evaluate 3. Get suggestions from people who are working on the tool

190. Describe some problem that you had with automating testing tool

1) The inability of win runner to identify the third party control like infragistics controls 2) The change of the location of the table object will cause object not found error. 3) The inability of the win runner to execute the script against multiple languages

191. What are the main attributes of test automation?

Maintainability, Reliability, Flexibility, Efficiency, Portability, Robustness, and Usability - these are the main attributes in test automation.

192. What testing activities you may want to automate in a project?

Testing tools can be used for : Sanity tests(which is repeated on every build), stress/Load tests(U simulate a large no of users, which is manually impossible) & Regression tests(which are done after every code change)

193. How to find that tools work well with your existing system?

To find this, select the suite of tests which are most important for your application. First run them with automated tool. Next subject the same tests to careful manual testing. If the results are coinciding you can say your testing tool has been performing.

194. How will you test the field that generates auto numbers of AUT when we click the button 'NEW" in the application?

We can create a text file in a certain location, and update the auto generated value each time we run the test and compare the currently generated value with the previous one will be one solution.

195. How will you evaluate the fields in the application under test using automation tool?

We can use Verification points (rational Robot) to validate the fields .Ex Using object data, object data properties VP we can validate fields.

196. Can we perform the test of single application at the same time using different tools on the same machine?

No. The Testing Tools will be in the ambiguity to determine which browser is opened by which tool.

197. What is 'configuration management'?

Configuration management is a process to control and document any changes made during the life of a project. Revision control, Change Control, and Release Control are important aspects of Configuration Management.

198. How to test the Web applications?

The basic difference in web testing is here we have to test for URL's coverage and links coverage. Using Win Runner we can conduct web testing. But we have to make sure that Web test option is

202 Software Testing Interview Questions and Answers

Take your testing career to the next level, get ISTQB certified – http://istqbexamcertification.com

selected in "Add in Manager". Using WR we cannot test XML objects.

199. What are the problems encountered during the testing the application compatibility on different browsers and on different operating systems

Font issues, alignment issues

200. How exactly the testing the application compatibility on different browsers and on different operating systems is done?

Compatibility testing is a type of software testing used to ensure compatibility of the system/application/website built with various other objects such as other web browsers, hardware platforms, users

201. How testing is proceeded when SRS or any other document is not given?

If SRS is not there we can perform Exploratory testing. In Exploratory testing the basic module is executed and depending on its results, the next plan is executed.

202. How do we test for severe memory leakages?

Severe memory leakages can be tested by using Endurance Testing . Endurance Testing means checking for memory leaks or other problems that may occur with prolonged execution.

# 2 Web Automation

## 2.1 Cucumber-JVM(Cucumber BDD)

### 2.1.1 What is Cucumber Annotations?

Annotation is a predefined text, which holds a selected which means. It lets the compiler/interpreter know, what need to be accomplished upon execution. Cucumber has given the following few annotations –

Given –

* It describes the pre-requisite for the test to be executed.
* Example − GIVEN I am a Facebook user.

When **−**

* It defines the trigger point for any test scenario execution.
* Example − WHEN I enter "<username>"

**Then −**

* Then holds the expected result for the test to be executed.
* Example − THEN login should be successful.

**And −**

* It presents the logical AND condition between any two statements. AND may be used along with GIVEN, whilst and then declaration.
* Example − WHEN I enter my "<username>" AND I enter my "<password>".

**But −**

* It indicates logical OR situation among any two statements. OR can be used together with GIVEN, when and then statement.
* Example − THEN login should be successful. BUT home page should not be missing.

**Scenario −**

* Details about the scenario under the test needs to be captured after the keyword “Scenario:”

**Example −**

Scenario:

GIVEN I am a Facebook user

WHEN I enter my

AND I enter my

THEN login should be successful.

BUT home page should not be missing.

* **Scenario Outline** − (To be covered later)
* **Examples** − (To be covered later)

Cucumber Tags:

There are mainly two varieties of tag –

**Default tag** − Default tag has their predefined meanings. example @Dev,@ignore

**Custom tag** − custom tag gives you complete flexibility to pick out suitable textual content for defining your tag.

Tag can also be defined at a function level. when you define a tag on the feature level, it ensures that every one the scenarios within that feature document inherits that tag. depending on the character of the scenario, we are able to use multiple tag for the single function. whenever Cucumber finds the proper call, a specific scenario will be done.

Cucumber also gives a way to inverse the choice of tags. remember that out of 25 described scenarios, 10 are marked as smoke check. we are required to execute best regression test scenarios.

What Is Cucumber Dry Run?  
Answer :  
  
Cucumber dry run is used to compile cucumber feature files and step Definitions. If there is any compilations errors it will show when we use dry run  
  
Ex: Cucumber features –dry-run  
  
What Is Scenario Outline?  
Answer :  
  
Scenario outline is used to execute the same scenario with different test data.

Is It Mandatory To Use The Keywords While Writing Scenario Steps ?  
Answer :  
  
No it is not mandatory to used keywords while writing scenario steps.  
  
We can write the scenario steps like the following without using keywords  
  
\* I am on the landed page  
  
How To Generate Cucumber Execution Reports?  
Answer :  
  
We can use the following command to generate html reports.  
  
–format html –out report.html –format pretty.  
  
  
   
  
How To Run A Particular Scenario From A Feature File ?  
Answer :  
  
We can run particular scenario from a feature file by giving the scenario line number  
  
Ex: cucumber features/test.feature:21  
  
What Is Cucumber And What Are The Advantages Of Cucumber?  
Answer :  
  
To run functional tests written in a plain text Cucumber tool is used. It is written in a Ruby programming language.  
  
Advantages of Cucumber:  
  
You can inolve business stakeholders who can not code  
End user experience is priority  
High code reuse  
What Are The 2 Files Required To Execute A Cucumber Test Scenario?  
Answer :  
  
The 2 files required to execute a Cucumber test scenario are  
  
Features  
Step Definition  
What Is Feature File In Cucumber? What Does Feature File Consist Of ?  
Answer :  
  
Feature file in cucumber consist of parameters or conditions required for executing code, they are:  
  
Feature  
Scenario  
Scenario Outline  
Given  
When  
Then  
Give An Example Of Behaviour Driven Test In Plain Text?  
Answer :  
  
Feature: Visit XYZ page in abc.com  
  
Scenario : Visit abc.com  
  
Given: I am on abc.com  
  
When: I click on XYZ page  
  
Then: I should see ABC page  
  
Explain What Is Scenario Outline In Feature File?  
Answer :  
  
Scenario Outline: Same scenario can be executed for multiple sets of data using scenario outline. The data is provided by a tabular structure separated by (I I).  
  
What Is Step Definition In Cucumber?  
Answer :  
  
A step definition is the actual code implementation of the feature mentioned in feature file.  
  
Give The Example For Step Definition Using “given” Function?  
Answer :  
  
For example to make visitor visit the site “wisdomjobs” the command we use for given:  
  
Given (/^ I am on www.wisdomjobs.com$/) do  
  
Browser.goto “http://www.wisdomjobs.com”  
  
end – This will visit www.wisdomjobs.com  
  
What Are The Difference Between Jbehave And Cucumber?  
Answer :  
  
Although Cucumber and Jbehave are meant for the same purpose, acceptance tests are completely different frameworks  
Jbehave is Java based and Cucumber is Ruby based  
Jbehave are based on stories while Cucumber is based on features.  
  
   
  
Explain What Is Test Harness?  
Answer :  
  
A test harness for cucumber and rspec allows for separating responsibility between setting up the context and interacting with the browser and cleaning up the step definition files.  
  
Explain When To Use Rspec And When To Use Cucumber?  
Answer :  
Rspec is used for Unit Testing.  
Cucumber is used behaviour driven development. Cucumber can be used for System and Integration Tests.  
What Is The Language Used For Expressing Scenario In Feature File ?  
Answer :  
Gherkin language is used to express scenario in feature files and ruby files containing unobtrusive automation for the steps in scenarios.  
  
Explain What Is Regular Expressions?  
Answer :  
A regular expression is a pattern describing a certain amount of text. The most basic regular expression consists of a single literal character.  
  
Explain What Is Bdd (behaviour Driven Development) ?  
Answer :  
BDD or Behaviour driven development is a process of developing software based on TDD (Test Driven Development) which focusses on behavioural specification of software units.  
  
What Softare Do You Need To Run A Cucumber Web Test?  
Answer :  
Ruby and its Development Kit  
Cucumber  
IDE like ActiveState  
Watir ( To simulate browser)  
Ansicon and rspec (if required)  
  
What Does A Features/ Support File Contains?  
Answer :  
Features/ support file contains supporting ruby code. Files in support load before those in step\_definitions, which can be useful for environment configuration.  
  
What Is Bdd Framework.what Is The Benefit Of Bdd In Selenium ?  
Answer :  
BDD is becoming widely accepted practice in agile software development, and Cucumber-JVM is a mainstream tool used to implement this practice in Java. Cucumber-JVM is based on Cucumber framework, widely used in Ruby on Rails world as well as in Java and .Net.  
  
Cucumber-JVM allows developers, QA, and non-technical or business participants to write features and scenarios in a plain text file using Gherkin language with minimal restrictions about grammar in a typical Given, When, and Then structure.  
  
The feature file is then supported by a step definition file, which implements automated steps to execute the scenarios written in a feature file. Apart from testing APIs with Cucumber-JVM, we can also test UI level tests by combining Selenium WebDriver.

## 2.2 Selenium Web Autmomation

### 2.2.1 How to write xpath:

1.

//tag[@attribute =’value’]

2.

//tag[@attribute=’value’ and @attribute =’value’]

3.

//tag[contains(text(), ‘text value’)]

4. Working on html table

//tag[text =’ text value’]//parent::td[@attribute=’value’]//preceding-sibling::td

[@attribute=’value’]//tag name[@attribute=’value’]

5.

//tag[text =’ text value’]//parent::td[@attribute=’value’]//preceding-sibling::td

[@attribute=’value’]//tag name[@attribute=’value’]

Selenium Exceptions

### 2.2.2 selenium.common.exceptions

**Exceptions that may happen in all the webdriver code.**

*exception* selenium.common.exceptions.**ElementClickInterceptedException**

(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ElementClickInterceptedException)

The Element Click command could not be completed because the element receiving the events is obscuring the element that was requested clicked.

*exception* selenium.common.exceptions.**ElementNotInteractableException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ElementNotInteractableException)

Thrown when an element is present in the DOM but interactions with that element will hit another element do to paint order

*exception* selenium.common.exceptions.**ElementNotSelectableException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ElementNotSelectableException)

**Thrown when trying to select an unselectable element.**

For example, selecting a ‘script’ element.

*exception* selenium.common.exceptions.**ElementNotVisibleException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ElementNotVisibleException)

Thrown when an element is **present on the DOM, but it is not visible**, and so is not able to be interacted with.

Most commonly encountered when trying to click or read text of an element that is hidden from view.

*exception* selenium.common.exceptions.**ErrorInResponseException**(*response*, *msg*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ErrorInResponseException)

Thrown when an **error has occurred on the server side.**

This may happen when communicating with the firefox extension or the remote driver server.

*exception* selenium.common.exceptions.**ImeActivationFailedException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ImeActivationFailedException)

Thrown when activating an IME engine has failed.

IME - stands for Input Method Engine. Currently seems like this is only supported in Linux platform and Firefox browser.

When working with Chinese/Japanese or multi-byte characters that needs to input by Selenium in linux, you have to use input framework like [IBus](https://github.com/ibus/ibus/wiki) and the engines implemented on IBus like [anthy](https://github.com/fujiwarat/ibus-anthy)(Japanese), [pinyin](https://github.com/phuang/ibus-pinyin/tree/master) (Chinese).

*exception* selenium.common.exceptions.**ImeNotAvailableException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ImeNotAvailableException)

Thrown when IME support is not available. This exception is thrown for every IME-related method call if IME support is not available on the machine.

*exception* selenium.common.exceptions.**InsecureCertificateException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InsecureCertificateException)

Navigation caused the user agent to hit a certificate warning, which is usually the result of an expired or invalid TLS certificate.

*exception* selenium.common.exceptions.**InvalidArgumentException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidArgumentException)

The arguments passed to a command are either invalid or malformed.

*exception* selenium.common.exceptions.**InvalidCookieDomainException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidCookieDomainException)

Thrown when attempting to add a cookie under a different domain than the current URL.

*exception* selenium.common.exceptions.**InvalidCoordinatesException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidCoordinatesException)

The coordinates provided to an interactions operation are invalid.

*exception* selenium.common.exceptions.**InvalidElementStateException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidElementStateException)

*exception* selenium.common.exceptions.**InvalidSelectorException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidSelectorException)

Thrown when the selector which is used to find an element does not return a WebElement. **Currently this only happens when the selector is an xpath expression** and it is either syntactically invalid (i.e. it is not a xpath expression) or the expression does not select WebElements (e.g. “count(//input)”).

*exception* selenium.common.exceptions.**InvalidSessionIdException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidSessionIdException)

Occurs if the given session id is not in the list of active sessions, meaning the session either does not exist or that it’s not active.

*exception* selenium.common.exceptions.**InvalidSwitchToTargetException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#InvalidSwitchToTargetException)

Thrown when frame or window target to be switched doesn’t exist.

*exception* selenium.common.exceptions.**JavascriptException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#JavascriptException)

An error occurred while executing JavaScript supplied by the user.

*exception* selenium.common.exceptions.**MoveTargetOutOfBoundsException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#MoveTargetOutOfBoundsException)

Thrown when the target provided to the ActionsChains move() method is invalid, i.e. out of document.

*exception* selenium.common.exceptions.**NoAlertPresentException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#NoAlertPresentException)

Thrown when switching to no presented alert.

This can be caused by calling an operation on the Alert() class when an alert is not yet on the screen.

*exception* selenium.common.exceptions.**NoSuchAttributeException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#NoSuchAttributeException)

Thrown when the attribute of element could not be found.

You may want to check if the attribute exists in the particular browser you are testing against. Some browsers may have different property names for the same property. (IE8’s .innerText vs. Firefox .textContent)

*exception* selenium.common.exceptions.**NoSuchCookieException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#NoSuchCookieException)

No cookie matching the given path name was found amongst the associated cookies of the current browsing context’s active document.

*exception* selenium.common.exceptions.**NoSuchElementException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#NoSuchElementException)

Thrown when element could not be found.

If you encounter this exception, you may want to check the following:

* Check your selector used in your find\_by...
* Element may not yet be on the screen at the time of the find operation, (webpage is still loading) see selenium.webdriver.support.wait.WebDriverWait() for how to write a wait wrapper to wait for an element to appear.

*exception* selenium.common.exceptions.**NoSuchFrameException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#NoSuchFrameException)

Thrown when frame target to be switched doesn’t exist.

*exception* selenium.common.exceptions.**NoSuchWindowException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#NoSuchWindowException)

Thrown when window target to be switched doesn’t exist.

To find the current set of active window handles, you can get a list of the active window handles in the following way:

**print** driver.window\_handles

*exception* selenium.common.exceptions.**RemoteDriverServerException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#RemoteDriverServerException)

*exception* selenium.common.exceptions.**ScreenshotException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#ScreenshotException)

A screen capture was made impossible.

*exception* selenium.common.exceptions.**SessionNotCreatedException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#SessionNotCreatedException)

A new session could not be created.

*exception* selenium.common.exceptions.**StaleElementReferenceException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#StaleElementReferenceException)

Thrown when a reference to an element is now “stale”.

Stale means the element no longer appears on the DOM of the page.

Possible causes of StaleElementReferenceException include, but not limited to:

* You are no longer on the same page, or the page may have refreshed since the element was located.
* The element may have been removed and re-added to the screen, since it was located. Such as an element being relocated. This can happen typically with a javascript framework when values are updated and the node is rebuilt.
* Element may have been inside an iframe or another context which was refreshed.

*exception* selenium.common.exceptions.**TimeoutException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#TimeoutException)

Thrown when a command does not complete in enough time.

*exception* selenium.common.exceptions.**UnableToSetCookieException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#UnableToSetCookieException)

Thrown when a driver fails to set a cookie.

*exception* selenium.common.exceptions.**UnexpectedAlertPresentException**(*msg=None*, *screen=None*, *stacktrace=None*, *alert\_text=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#UnexpectedAlertPresentException)

Thrown when an unexpected alert is appeared.

Usually raised when when an expected modal is blocking webdriver form executing any more commands.

*exception* selenium.common.exceptions.**UnexpectedTagNameException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#UnexpectedTagNameException)

Thrown when a support class did not get an expected web element.

*exception* selenium.common.exceptions.**UnknownMethodException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#UnknownMethodException)

The requested command matched a known URL but did not match an method for that URL.

*exception* selenium.common.exceptions.**WebDriverException**(*msg=None*, *screen=None*, *stacktrace=None*)[[source]](https://seleniumhq.github.io/selenium/docs/api/py/_modules/selenium/common/exceptions.html#WebDriverException)

Base webdriver exception.

### 2.2.3 What are difference between Selenium IDE, RC and WebDriver

|  |  |  |
| --- | --- | --- |
| **Selenium IDE** | **Selenium RC** | **Selenium WebDriver** |
| It only works in Mozilla browser. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. |
| It supports Record and playback | It doesn’t supports Record and playback | It doesn’t supports Record and playback |
| Doesn’t required to start server before executing the test script. | Required to start server before executing the test script. | Doesn’t required to start server before executing the test script. |
| It is a GUI Plug-in | It is standalone java program which allow you to run Html test suites. | It actual core API which has binding in a range of languages. |
| Core engine is Javascript based | Core engine is Javascript based | Interacts natively with browser application |
| Very simple to use as it is record & playback. | It is easy and small API | As compared to RC, it is bit complex and large API. |
| It is not object oriented | API’s are less Object oriented | API’s are entirely Object oriented |
| It doesn’t supports of moving mouse cursors. | It doesn’t supports of moving mouse cursors. | It supports of moving mouse cursors. |
| Need to append full xpath with ‘xpath=\\’ syntax | Need to append full xpath with ‘xpath=\\’ syntax | No need to append full xpath with ‘xpath=\\’ syntax |
| It does not supports listeners | It does not supports listeners | It supports the implementation of listeners |
| It does not support to test iphone/Android applications | It does not support to test iphone/Android applications | It support to test iphone/Android applications |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 2.2.4 Synchronization in Selenium Webdriver

<http://www.seleniumeasy.com/selenium-tutorials/synchronization-in-selenium-webdriver>

It is a mechanism which involves more than one components to work parallel with Each other.

Generally in Test Automation, we have two components

2.0.2.1 Application Under Test

2.0.2.2 Test Automation Tool.

Both these components will have their own speed. We should write our scripts in such a way that both the components should move with same and desired speed, so that we will not encounter "Element Not Found" errors which will consume time again in debugging.

Synchronization can be classified into two categories:

2.0.2.1 Unconditional

2.0.2.2 Conditional Synchronization

#### 2.2.4.1 Unconditional :

In this we just specify timeout value only. We will make the tool to wait until certain amount of time and then proceed further.

*Examples: Wait() and [Thread.Sleep();](https://docs.oracle.com/javase/tutorial/essential/concurrency/sleep.html)*

The main disadvantage for the above statements are, there is a chance of unnecessary waiting time even though the application is ready.

The advantages are like in a situation where we interact for third party systems like interfaces, it is not possible to write a condition or check for a condition. Here in this situations, we have to make the application to wait for certain amount of time by specifying the timeout value.

#### 2.2.4.2 Conditional Synchronization:

We specify a condition along with timeout value, so that tool waits to check for the condition and then come out if nothing happens.

It is very important to set the timeout value in conditional synchronization, because the tool should proceed further instead of making the tool to wait for a particular condition to satisfy.

In Selenium we have implicit Wait and Explicit Wait conditional statements. Check here for [Examples on how to use Webdriver Waits](http://seleniumeasy.com/selenium-tutorials/webdriver-wait-examples)

##### 2.2.4.2.1 Implicit Wait.

An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available.

The default setting is 0. Once when we define the implicit wait, it will set for the life of the WebDriver object instance.

It is a mechanism which will be written once and applied for entire session automatically. It should be applied immediately once we initiate the Webdriver.

Implicit wait will not work all the commands/statements in the application. It will work only for "FindElement" and "FindElements" statements.

If we set implicit wait, find element will not throw an exception if the element is not found in first instance, instead it will poll for the element until the timeout and then proceeds further. We should always remember to add the below syntax immediately below the Webdriver statement.

Syntax:

driver.manage.TimeOuts.implicitwait(6,Timeunit.SECONDS);

**Example using implicit timeout**

WebDriver driver = **new** FirefoxDriver();  
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
driver.**get**("[http://www.google.com"](http://www.google.com/));

##### 2.2.4.2.2 Explicit Wait:

We need to define a wait statement for certain condition to be satisfied until the specified timeout period. If the Webdriver finds the element within the timeout period the code will get executed.

Explicit wait is mostly used when we need to Wait for a specific content/attribute change after performing any action, like when application gives AJAX call to system and get dynamic data and render on UI.

Example: Like there are drop-downs Country and State, based on the country value selected, the values in the state drop-down will change, which will take few seconds of time to get the data based on user selection.

Example:

/\*Explicit **wait** **for** **state** dropdown field\*/  
 WebDriverWait **wait** = new WebDriverWait(driver, 10);  
 **wait**.**until**(ExpectedConditions.visibilityOfElementLocated(By.id("statedropdown")));

The above statement waits up to 10 seconds before throwing Exception (TimeoutException - Timed out after 10 seconds waiting for visibility of element) or if it finds the element, it will return in 0 - 10 seconds.

There are different waits that can be used based on the needs which we frequently come across when automating web applications. Check here for [WebDriver Waits Example](http://seleniumeasy.com/selenium-tutorials/webdriver-wait-examples).

##### 2.2.4.2.3 Fluent Wait:

Using FluentWait we can define the maximum amount of time to wait for a condition, as well as the frequency with which to check for the condition.

And also the user can configure to ignore specific types of exceptions such as ["NoSuchElementExceptions"](http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/NoSuchElementException.html)when searching for an element. NoSuchElement exception is thrown by findElement(By) and findElements(By). Whenever it try to find any element it returns the first matching element on the current page else it throws NoSuchElementException - when no matching elements are found.

Syntax:

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)  
 //Wait for the condition  
 .withTimeout(30, TimeUnit.SECONDS)   
 // which to check for the condition with interval of 5 seconds.   
 .pollingEvery(5, TimeUnit.SECONDS)   
 //Which will ignore the NoSuchElementException  
 .ignoring(NoSuchElementException.class);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 2.2.5 What are the ways to refresh a browser using Selenium WebDriver?

There are multiple ways to refresh a page in selenium

* Using *driver.navigate().refresh()* command as mentioned in the question 45
* Using driver.get(“URL”) on the current URL or using *driver.getCurrentUrl()*
* Using driver.navigate().to(“URL”) on the current URL or *driver.navigate().to(driver.getCurrentUrl());*
* Using *sendKeys(Keys.F5)* on any textbox on the webpage

### 2.2.6 What is the difference between driver.close() and driver.quit() methods?

Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

*driver.close():* To close current WebDriver instance

*driver.quit():* To close all the opened WebDriver instances

### 2.2.7 What is the difference between driver.findElement() and driver.findElements() commands?

The difference between driver.findElement() and driver.findElements() commands is-

* findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas **findElements() returns a list** of WebElements, all satisfying the locator value passed.
* Syntax of findElement()-
* WebElement textbox = driver.findElement(By.id(“textBoxLocator”));
* Syntax of findElements()-
* List <WebElement> elements = element.findElements(By.id(“value”));
* **Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.**

### 2.2.8 How to capture Screenshot in Selenium WebDriver?

By using *TakesScreenshot* Interface

In Selenium 3, we may face few issues while capturing Screenshots. To overcome we use aShot utility. Click on below links to see posts related to the normal way of capturing a screenshot and capturing a screenshot using aShot utility.

[Capture screenshot using Selenium WebDriver](https://www.softwaretestingmaterial.com/capture-screenshot-using-selenium-webdriver/)

[Full Page Screenshot using aShot utility](https://www.softwaretestingmaterial.com/how-to-capture-full-page-screenshot-using-selenium-webdriver/)

[Failed Test Cases Screenshot](https://www.softwaretestingmaterial.com/capture-screenshot-of-failed-test-cases-using-selenium-webdriver/)

### 2.2.9 Bitmap checkpoint in Selenium

♠ Posted by Testing World at 3:44 AM

**Compare Image Files Using Selenium**

**Step 1:** Import Java.io.\* package

We need to create File type of object, where we will open jpg file.

**Step 2:** Create a file object for both input files.

File f1 = **new** File("C:\\File\\168022.JPG");

File f2 = **new** File("C:\\File\\168024.JPG");

**Step 3:** Import java.awt.image.\* package

**Step 4:** Create a **buferedImage** object and pass file object in that

File f1 = **new** File("C:\\File\\168022.JPG");

File f2 = **new** File("C:\\File\\168024.JPG");

BufferedImage b1 = ImageIO.*read*(f1);

BufferedImage b2 = ImageIO.*read*(f2);

**Step 5 :** For the high level of testing, we can compare height and width of both image

Use **getWidth** and **getHeight** method of **bufferedImage** object

file1\_height=b1.getHeight();

file1\_width=b1.getWidth();

file2\_height=b2.getHeight();

file2\_width=b2.getWidth();

**if**(file1\_height==file2\_height && file2\_width==file1\_width)

{

System.*out*.println("File dimenstions are same");

}

**Step 6:** Now we can find RGB value of each pixel of the file and compare with the each pixel value of other file

getRGB will return an integer reflect t particular RGB value

**if**(file1\_height==file2\_height && file2\_width==file1\_width)

{

**int** i,j;

**for**(i=1;i<=file1\_height;i++)

{

**for**(j=1;j<=file1\_width;j++)

{

**if**(b1.getRGB(i, j)!= b2.getRGB(i, j))

{

System.*out*.println("NOT SAME");

**break**;

}

}

}

**package** org.nitin;

**import** java.io.\*;

**import** java.awt.image.\*;

**import** javax.imageio.ImageIO;

**public** **class** LearnString {

**public** **static** **void** main(String aa[])

{

**int** file1\_height, file1\_width,file2\_height, file2\_width;

**try**

{

File f1 = **new** File("C:\\File\\168022.JPG");

File f2 = **new** File("C:\\File\\168023.JPG");

BufferedImage b1 = ImageIO.*read*(f1);

BufferedImage b2 = ImageIO.*read*(f2);

file1\_height=b1.getHeight();

file1\_width=b1.getWidth();

file2\_height=b2.getHeight();

file2\_width=b2.getWidth();

**if**(file1\_height==file2\_height && file2\_width==file1\_width)

{

**int** i,j;

**for**(i=1;i<=file1\_height;i++)

{

**for**(j=1;j<=file1\_width;j++)

{

**if**(b1.getRGB(i, j)!= b2.getRGB(i, j))

{

System.*out*.println("NOT SAME");

**break**;

}

}

}

}

} **catch**(Exception ex){}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 2.2.10 How can we handle windows based pop up?

Selenium doesn’t support windows based applications. It is an automation testing tool which supports only web application testing. We could handle windows based popups in Selenium using some third party tools such as AutoIT, Robot class etc.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 2.2.11 Upload file AutoIt Script in Selenium WebDriver:

If there is no text box to set the file path and only able to click on Browse button to upload the file in the windows popup box then we do upload file using *AutoIt* tool.

**AutoIt Introduction:**

*AutoIt* Tool is an open source tool. It is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g. [VBScript](https://www.softwaretestingmaterial.com/vbscript-for-automation-qtp-uft-testing/) and SendKeys). *AutoIt* is also very small, self-contained and will run on all versions of Windows out-of-the-box with no annoying “runtimes” required!

Now the question is how we do upload file using AutoIT Tool in Selenium WebDriver.

**Follow the below steps:**

1. Download [*Autoit*](http://www.autoitscript.com/site/autoit/) tool from here and install it
2. Open Programs – *Autoit* tool – SciTE Script Editor and add the below mentioned *AutoIt* script in *Autoit* editor and save it as ‘UploadFile.au3’ in your system
3. Convert it as ‘UploadFile.exe’
4. In Eclipse, add the below mentioned Selenium Script and run

**Step 1:** Download [*AutoIt*](http://www.autoitscript.com/site/autoit/) *tool and install*

**Step 2:** Open SciTE Script editor and add the below mentioned *AutoIt script and s*ave it as ‘UploadFile.au3’ in your system.

***AutoIt S*cript:**

|  |  |
| --- | --- |
| 1  2  3 | WinWaitActive("File Upload")  Send("D:\SoftwareTestingMaterial\UploadFile.txt")  Send("{ENTER}") |

***AutoIt S*cript Explanation:**

Line 1 : *WinWaitActive(“File Upload”)*

Above line of code changes the focus of cursor on the Window popup box to upload file.

**‘*File Upload*‘** is the name of the window popup when using **Mozilla Firefox.** If you want to use other browsers such as **Chrome you need to pass the value as ‘*Open*‘ (‘Open’ is the name of the window popup**) and for IE you need to pass the value as ‘File To Upload’ (‘File To Upload’ is the name of the window popup)

Line 2 : *Send(“Path of the document”)*

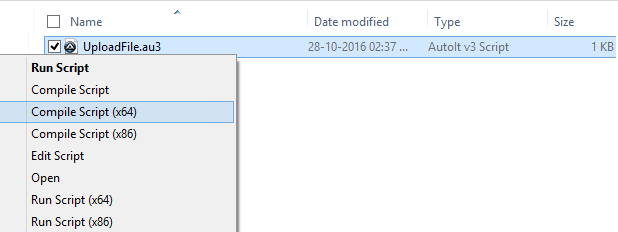
Once the window popup is active, it sets the path of the document which needs to be uploaded

*Send(“D:\SoftwareTestingMaterial\UploadFile.txt”)*

Line 3 : *Send(“{ENTER}”)*

After that it clicks on Open button which will upload the document

**Step 3:** Once the file is saved, we need to convert the ‘UploadFile.au3’ to ‘UploadFile.exe’. To do this we need to compile the ‘UploadFile.au3’



Right click on the file ‘UploadFile.au3’ and click on ‘Compile Script’ to generate an executable file ‘UploadFile.exe’

**Step 4:** In Eclipse, add the below mentioned Selenium Script and run

Given clear explanation in the comments section with in the program itself. Please go through it to understand the flow.In the above Selenium Script, we did call the *AutoIt* Script after clicking on the browser button which transfers windows popup box and upload the required file.

**Syntax:**

|  |  |
| --- | --- |
| 1 | Runtime.getRuntime().exec("File Path of AutoIt.exe"); |

*Runtime.getRuntime().exec(“D:\\SoftwareTestingMaterial\\AutoIt\\Uploadfile.exe”);*

This way we could upload a file using AutoIT.

In short For my hybrid Framework

*Used AutoIt to upload file from Windows.*

*1.Create FileUpload.au3 in ‘SciTe’ with bellow line of codes*

*(WinWaitActive("Open")*

*//Open for chrome browser in windows*

*Send("C:\Users\Rabeka\IdeaProjects\Team\_3\_Framework\_Project\AliBaba\data\autoItFiles\laptop.jpg")*

*Send("{ENTER}")*

*2.Convert FileUpload.au3 to FileUpload.exe file by AutExe converter*

*Inside the helper method use this syntax to uoload a file from windows machine.*

*Runtime.getRuntime().exec(“D:\\SoftwareTestingMaterial\\AutoIt\\Uploadfile.exe”);*

How to download file by Selenium WebDriver:

*1.Create FileDownload.au3 in ‘SciTe’ with bellow line of codes*

; wait for 8 seconds to appear download and save dialog. Used class property of download dialog.

WinWait("[CLASS:#MozillaDialogClass]","",8)

; Perform keyboard ALT key + s key to select Save File Radio button using keyboard shortcut.

Send("!s")

; Wait for 9 seconds

Sleep(9000)

; Press Keyboard ENTER button.

Send("{ENTER}")

*2.Convert FileUpload.au3 to FileUpload.exe file by AutExe converter*

*3.Inside the helper method use this syntax to download a file from windows machine.*

*Runtime.getRuntime().exec(“D:\\SoftwareTestingMaterial\\AutoIt\\DownLoadFile.exe”);*

[*https://www.softwaretestingmaterial.com/download-file-using-autoit/*](https://www.softwaretestingmaterial.com/download-file-using-autoit/)

How to resize window?

//Create object of Dimensions class

Dimension d = new Dimension(480,620);

//Resize the current window to the given dimension

driver.manage().window().setSize(d);

Scroll to the top of the page?

//to perform Scroll on application using Selenium

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,document.body.scrollHeight)");

Scroll anywhere in the page?  
//to perform Scroll on application using Selenium

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,250)", "");

How To Perform Right Click Action (Context Click) In Selenium WebDriver?

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

[Practical Example](https://www.softwaretestingmaterial.com/selenium-right-click-action/)

### 2.2.12 How To Perform Right Click Action (Context Click) In Selenium WebDriver?

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

//Create the object 'action'

Actions action = new Actions(driver);

//contextClick() method to do right click on the element

action.contextClick(rightClickElement).build().perform();

### 2.2.13 How to perform drag and drop?

//To get source locator

WebElement sourceLocator = driver.findElement(By.xpath("xpath"));

//To get target locator

WebElement targetLocator = driver.findElement(By.xpath("xpath"));

//create object 'action' of Actions class

Actions action = new Actions(driver);

//use dragAndDrop() method. It accepts two parametes source and target.

action.dragAndDrop(sourceLocator, targetLocator).build().perform();

#### 2.2.13.1 Selenium Actions Class:

Selenium has the built-in ability to handle various types of keyboard and mouse events. In order to do action events, you need to use*org.openqa.selenium.interactions* Actions class. The user-facing API for emulating complex user gestures. Use the selenium actions class rather than using the Keyboard or Mouse directly. This API includes actions such as drag and drop, clicking multiple elements.

To create an object ‘*action*‘ of Selenium Actions class:

|  |  |
| --- | --- |
| 1 | Actions action=new Actions(driver); |

To focus on element using WebDriver:

|  |  |
| --- | --- |
| 1 | action.moveToElement(element).perform(); |

*element* is the webelement which we capture

*perform()* method is used here to execute the action.

To click on the element:

|  |  |
| --- | --- |
| 1 | action.moveToElement(element).click().perform(); |

*click()* method is used here to click the element.

#### 2.2.13.2 Methods Available in Selenium Actions Class:

Keyboard Events Using Selenium Actions Class API:

The Keyboard interface has the below mentioned methods:

* sendKeys(keysToSend) : sends a series of keystrokes onto the element
* keyDown(theKey) : Sends a key press without release it. Subsequent actions may assume it as pressed. (example: Keys.ALT, Keys.SHIFT, or Keys.CONTROL)
* keyUp(theKey): Performs a key release

#### 2.2.13.3 Mouse Events Using Selenium Actions Class API:

* click (): Simply click on element
* doubleClick (): Double clicks onElement
* contextClick() : Performs a context-click (right click) on an element
* clickAndHold(): Clicks at the present mouse location (without releasing)
* dragAndDrop(source, target): Invokes click-and-hold at the source location and moves to the location of the target element before releasing the mouse. source – element to grab, target – element to release
* dragAndDropBy(source, xOffset, yOffset) : Performs click-and-hold at the source location, shifts by a given offset, then frees the mouse. xOffset – to shift horizontally, yOffset – to shift vertically
* moveByOffset(x-offset, y-offset): Shifts the mouse from its current position (or 0,0) by the given offset. x-offset – Sets the horizontal offset (negative value – shifting the mouse to the left), y-offset – Sets the vertical offset (negative value – shifting the mouse to the up)
* moveToElement(toElement): It shifts the mouse to the center of the element
* release(): Releases the depressed left mouse button at the existing mouse location

Some scenarios where we use Selenium Actions class are mentioned below. Check out the below links.

[How to handle mouse hover actions using Actions in Selenium](https://www.softwaretestingmaterial.com/mouse-hover-actions-using-selenium/)

[How to do Drag and Drop using Actions in Selenium](https://www.softwaretestingmaterial.com/drag-and-drop-using-actions-class-in-selenium/)

[How to Scroll Web Page Down or Up Using Selenium](https://www.softwaretestingmaterial.com/scroll-web-page-using-actions-class/)

[How to Perform Context Click / Right Click using Actions in Selenium](https://www.softwaretestingmaterial.com/selenium-right-click-action/)

[How to Perform Double Click using Actions in Selenium](https://www.softwaretestingmaterial.com/double-click-action-selenium/)

**Website Not Using Authentication:** If a website is only using the HTTP and does not require any authen- tication, then it’s very easy to handle in Selenium.

*WebDriver driver = new ChromeDriver();*

*driver.get("http://destination-url/");*

**Website Using Basic Authentication:** There are many websites which apply basic authentication scheme be- fore allowing access to their home page. Here are three most prominent ways to do it in Selenium. Below code is using WebDriverWait and Alert classes to implement basic authentication for HTTP.

*WebDriverWait testwait = new WebDriverWait(driver, 10);   
Alert testalert =testwait.until(ExpectedConditions.alertIsPresent()); testalert.authenticateUsing(new UserAndPassword(userName,*password)); Alternatively, we can also pass the user/pass pair within the HTTP URL as a parameter to the Webdriver’s Get method. String target = “http://user:pass@host”;

### 2.2.14 Handle HTTPS Protocol:

HTTPS is the secure version of HTTP. And it encrypts all the communication exchanged between the web server and the browser. It makes use of SSL certificates which get downloaded into the browser upon initiating the first request. The certificate contains a **public key for encrypting** the data flowing from client to the server. Similarly, the server reserves a **private key** corresponding to the certificate and use it to decrypt the data received. These public/private keys are none other than the famous RSA key- pair and available in different key lengths such as 512, 1024, 2048, and 4096 bits. So in reality, we need to handle the SSL certificates via Selenium. Also, we’ve to follow different approaches for managing the SSL in Firefox, Chrome, and IE browsers.

**RSA:**

RSA is an [algorithm](https://simple.wikipedia.org/wiki/Algorithm) used by modern computers **to** [**encrypt**](https://simple.wikipedia.org/wiki/Encryption) **and decrypt messages**. It is an asymmetric **cryptographic algorithm**. Asymmetric means that there are two different [keys](https://simple.wikipedia.org/wiki/Key_(cryptography)). This is also called [public key cryptography](https://en.wikipedia.org/wiki/Public-key_cryptography), because one of them can be given to everyone. The other key must be kept private.RSA stands for [**Ron** Rivest](https://en.wikipedia.org/wiki/Ron_Rivest), [Adi **Shamir**](https://en.wikipedia.org/wiki/Adi_Shamir) and [Leonard **Adleman**](https://en.wikipedia.org/wiki/Leonard_Adleman), who first publicly described RSA in 1978. RSA involves a **public** [**key**](https://simple.wikipedia.org/wiki/Key_(cryptography)) **and private key**. The public key can be known to everyone; it is used to encrypt messages. Messages encrypted using the public key can only be decrypted with the private key.

### 2.2.15 What is Parameterized testing in TestNG?

*Parameterized tests* allow developers to run the same test over and over again using different values.

There are two ways to set these parameters:

* *with testng.xml -* [Practical Example](https://www.softwaretestingmaterial.com/testng-parameterization-using-xml/)
* *with Data Providers –* [Practical Example](https://www.softwaretestingmaterial.com/testng-parameterization-using-dataproviders/)

In our hybrid framework we use heavily parameterized testing. We pass browser information, version, windows version, url, Operating System name, environment name whether it is cloud or local etc under parameter tag and @Parameter annotations in generic received those parameter accordingly.

### 2.2.16 How to run a group of test cases using TestNG?

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set. This gives you maximum flexibility in how you partition your tests and doesn’t require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your testng.xml file and can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.

|  |  |
| --- | --- |
| 1  2  3  4 | @Test (groups = { "smokeTest", "functionalTest" })  public void loginTest(){  System.out.println("Logged in successfully");  } |

<suite name="softwaretestingmaterial">

<test name="testngTest">

<groups>

<run>

<include name="smokeTest" />

</run>

</groups>

<classes>

<class name="softwareTestingMaterial.TestCase1" />

<class name="softwareTestingMaterial.TestCase2" />

</classes>

</test>

</suite>

### 2.2.17 What is the use of @Test(invocationCount=x)?

The *invocationcount* attribute tells how many times TestNG should run a test method

|  |  |
| --- | --- |
| 1  2 | @Test(invocationCount = 10)  public void testCase1(){ |

In this example, the method *testCase1* will be invoked ten times

### What are some expected conditions that can be used in Explicit waits?

**Answer:** Some of the commonly used expected conditions of an element that can be used with expicit waits are-

•elementToBeClickable(WebElement element or By locator)

stalenessOf(WebElement element)

•visibilityOf(WebElement element)

•visibilityOfElementLocated(By locator)

•invisibilityOfElementLocated(By locator)

•attributeContains(WebElement element, String attribute, String value)

•alertIsPresent()

•titleContains(String title)

•titleIs(String title)

•textToBePresentInElementLocated(By, String)

### 2.2.18 What are some commonly encountered exceptions in selenium?

Answer:

Some of the commonly seen exception in selenium are-

•NoSuchElementException - When no element could be located from

the locator provided.

•ElementNotVisibleException - When element is present in the dom

but is not visible.

•NoAlertPresentException - When we try to switch to an alert but the

targetted alert is not present.

NoSuchFrameException - When we try to switch to a frame but the

targetted frame is not present.

•NoSuchWindowException - When we try to switch to a window but

the targetted window is not present.

•UnexpectedAlertPresentException - When an unexpected alert blocks

normal interaction of the driver.

•TimeoutException - When a command execution gets timeout.

•InvalidElementStateException - When the state of an element is not

appropriate for the desired action.

•NoSuchAttributeException - When we are trying to fetch an attribute's

value but the attribute is not correct

•WebDriverException - When there is some issue with driver instance

preventing it from getting launched.

------------------------------

### 2.2.19 Which version of selenium you are using now?

Answer: 3.7.1 (Hybrid Framework)

### 2.2.20 What are the changes available in Selenium 3?

The major difference between Selenium 2 and Selenium 3 is, Selenium 3 has bug fixes from selenium 2. Also, **Selenium 3 is more mobile automation focused.**

Here is the summary of the change from Official Selenium Blog.

* For WebDriver users, it's more of bug fixes and drop-in replacement for 2.x
* Selenium Grid bug fixes are done as well.
* Selenium project will not actively support only the WebDriver API.
* The **Selenium RC APIs have been moved to a “legacy” package.**
* The original code powering Selenium RC has been replaced with something backed by WebDriver, which is also contained in the "legacy" package.
* Mozilla has made internals of Firefox browser more stable. So community provided Firefox Driver will no longer work. That means that from Firefox 48 you must use their geckodriver to use that browser, regardless of whether you're using Selenium 2 or 3.

https://raw.githubusercontent.com/SeleniumHQ/selenium/master/java/CHANGELOG

### 2.2.21 How did you create firefox driver in Selenium 2 and How do you create Firefox driver in Selenium 3?

Selenium Webdriver 3.0

System.setProperty("webdriver.gecko.driver","path of geckodriver.exe");  
WebDriver driver = **new** FirefoxDriver();

Selenium Webdriver 2.xx

WebDriver driver = **new** FirefoxDriver();

### 2.2.22 What are the difficulties you may face while using Webdriver in IE browser ?

As we all know, InternetExplorerDriver works only with Windows system and the execution speed is slow Comparatively to other browsers.

#### 2.2.22.1 NoSuchElementFound

Most of the time when working with Internet explorer, we may end up seeing issues such as 'NoSuchElementFound' exception because of [Synchronization](http://seleniumeasy.com/selenium-tutorials/synchronization-in-selenium-webdriver)

#### 2.2.22.2 Issues with mouse events

However when working with InternetExplorerDriver there are some issues with **mouse events when the browser window does not have focus**, and attempting to hover over elements.

Your test scripts may work fine with Firefox and Chrome browsers which are intelligent enough find the elements in the DOM, but Internet Explorer is slow because of which you will end up with an exception.

To avoid issues when executing scripts with Internet explorer, try to use ['Css selectors'](http://seleniumeasy.com/selenium-tutorials/css-selectors-tutorial-for-selenium-with-examples) which will minimize your issues.

### 2.2.23 When ever working with Internet explorer browser for Selenium webdriver, the below are the common issues that you may come across.

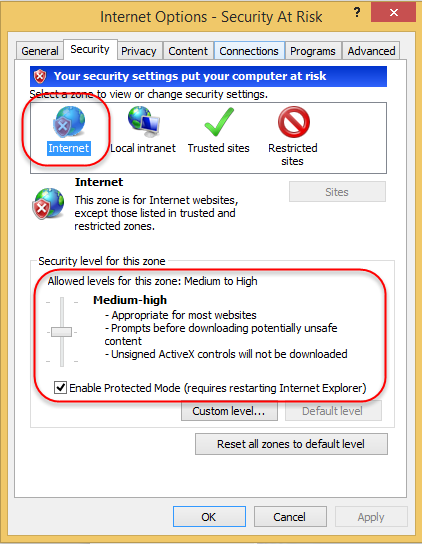
#### 2.2.23.1 If see issue some thing like 'Unexpected error launching Internet Explorer' below, You have to set 'Enable protected mode' option in all levels with same value.

org.openqa.selenium.remote.SessionNotFoundException: Unexpected error launching Internet Explorer. Protected Mode settings are not the same for all zones. Enable Protected Mode must be **set** **to** the same **value** (enabled **or** disabled) **for** **all** zones. (WARNING: The server did **not** provide **any** stacktrace information)  
Command duration **or** timeout: 516 milliseconds

**Please follow below steps to set:**

1. Open Internet Explorer browser--> Select Internet Options from Tools menu

2. Select Security Tab --> Select Enable Protected Mode option -- > Check the default Zone level for 'Internet'. If you look at the screen shot below, security level for this zone is selected as 'Allowed level for this zone : Medium to High.' and 'Enable Protected Mode' option is Checked.



Now you need to make sure that, for the other Zones, such as 'Local Internet' and 'Trusted sites' is also selected as ABOVE. You may don't need to do anything with 'Restricted Site' option. We can leave the option as is and by default 'Enable Protected Mode' option will be Checked.

Now after changing the settings, please click on 'Apply' and 'Ok' button.

There is also an other alternative for setting the protected mode using desired capabilities as below: -

DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();  
capabilities.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,true);

But however, the first option is advised and it is not that hard to set internet explorer browser settings.

You can checkout for the required configuration options which are defined [IE required configuration for Selenium Webdriver](https://code.google.com/p/selenium/wiki/InternetExplorerDriver#Required_Configuration)

#### 2.2.23.2 Make sure that the IE browser zoom level is set to 100% so that the native mouse events can be set to the correct coordinates.

#### 2.2.23.3 It may be silly one, But make sure you provide correct path when setting the property of Internet explorer driver.

### 2.2.24 Element is not clickable at point SeleniumWebdriverException?

<http://www.seleniumeasy.com/selenium-tutorials/element-is-not-clickable-at-point-selenium-webdriver-exception>

This issue comes only when working with chrome driver as the chrome browsers uses point location. When the element position is not fixed and we are trying to do some action on that particular element will result an error as 'selenium.common.exceptions.WebDriverException - Element is not clickable at point (xx, xx). Other element would receive the click'.

This happens when the element is loaded into the DOM, but the position is not fixed on the UI. There can be some other divs or images or ads that are not loaded completely. And ChromeDriver always tries to click the middle of the element.

There are work around that worked to resolve the issue. But to make sure the best and simple solution is to find out the exact reason to fix the problem. We need to figure out which part of the div/image is taking time to load. Before clicking on an element we need to make sure the element is present in the DOM, visible in the UI and the last is Position is fixed. When the element position is fixed the problem is solved. If you want to check that, try Thread.sleep or verify in debug mode.

But using Thread.sleep is not at all a good idea. Instead we should go for [WebDriverWait ExpectedConditions](https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html).

There are many Conditions that we can use within Webdriver tests.

1. **visibilityOf(WebElement element) :** An expectation for checking that an element, known to be present on the DOM of a page, is visible.

2. **visibilityOfElementLocated(By locator) :** An expectation for checking that an element is present on the DOM of a page and visible.

In the above two conditions we are waiting for an element to be present on the DOM of a page and also visible. These works fine only when the element is loaded completely.

Even though the element is loaded and visible. For chrome driver the position of the element is also important. Remember this will work perfectly on Firefox driver.

Then How to solve this problem in Chrome????? . You can also check the number of comments and discussions on [Chrome Element is not clickable issue](https://code.google.com/p/selenium/issues/detail?id=2766)

**The below are the different work around that worked for people who also faced the same problem.**

1. This is a simple solution that worked for many people. Try to maximize the browser when you are working with resolutions greater than 1024x768.

driver.manage().**window**().maximize();

2. The other solution which also worked for few users using Actions Class

WebElement element = driver.findElement(By("element"));  
Actions action = **new** Actions(driver);  
action.moveToElement(element).click().perform();

3. It worked for some using JavaScriptExecutor. Make sure to import org.openqa.selenium.JavascriptExecutor;

JavascriptExecutor js = (JavascriptExecutor)driver;  
 // **if** the element **is** on top.  
js.executeScript("scroll(250, 0)");  
  
// **if** the element **is** on bottom.  
js.executeScript("scroll(0, 250)");

4. You can also try the below using X or Y position

WebElement element = driver.findElement(By.id(""));  
 JavascriptExecutor js =(JavascriptExecutor)driver;  
 js.executeScript("window.scrollTo(0,"element.getLocation().x+")");  
 element.click();

Or

WebElement element = driver.findElement(By.id(""));  
 JavascriptExecutor js =(JavascriptExecutor)driver;  
 js.executeScript("window.scrollTo(0,"element.getLocation().y+")");  
 element.click();

### 2.2.25 Different examples to work with Alerts, Windows, Frames and Authentication Window

#### [2.2.25.1 Working with Frames in Selenium Webdriver](http://seleniumeasy.com/selenium-tutorials/how-to-work-with-iframes-in-selenium-webdriver)

Three way to catch Frame:

1. Index Number(ID)
2. Frame Name
3. WebElement from the frame.

What is iFrame?

An iFrame (Inline Frame) is an HTML document embedded inside the current HTML document on a website. iFrame HTML element is used to insert content from another source, such as an advertisement, into a Web page. A Web designer can change an iFrame's content without making them reload the complete website. A website can have multiple frames on a single page. And a frame can also have inner frames (Frame in side a Frame)

---------------------------------------  
| | |  
| | |  
| Frame 1 | |  
| | |  
| | |  
|--------------| |  
| | Frame 3 |  
| | |  
| | |  
| | |  
| Frame 2 | |  
| | |  
| | |  
| | |  
| | |  
 --------------------------------------------

In Selenium to work with iFrames, we have different ways to handle frame depending on the need. Please look at the below ways of handling frames

**driver.switchTo().frame(int arg0);**

Select a frame by its (zero-based) index. That is, if a page has multiple frames (more than 1), the first frame would be at index "0", the second at index "1" and so on.

Once the frame is selected or navigated , all subsequent calls on the WebDriver interface are made to that frame. i.e the driver focus will be now on the frame. What ever operations we try to perform on pages will not work and throws element not found as we navigated / switched to Frame.

*Parameters: Index - (zero-based) index*

*Returns: driver focused on the given frame (current frame)*

*Throws: NoSuchFrameException - If the frame is not found.*

*Example: if iframe id=webklipper-publisher-widget-container-frame*, it can be written as driver.switchTo().frame("webklipper-publisher-widget-container-frame"); Below is the code snippet to work with switchToFrame using frame id.

|  |
| --- |
| **public** **void** **switchToFrame**(**int** frame) {  **try** {  driver.switchTo().frame(frame);  System.out.println("Navigated to frame with id " + frame);  } **catch** (NoSuchFrameException e) {  System.out.println("Unable to locate frame with id " + frame  + e.getStackTrace());  } **catch** (Exception e) {  System.out.println("Unable to navigate to frame with id " + frame  + e.getStackTrace());  }  } |

**driver.switchTo().frame(String arg0);**

Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.

*Parameters: name Or Id - the name of the frame or the id of the frame element.*

*Returns: driver focused on the given frame (current frame)*

*Throws: NoSuchFrameException - If the frame is not found*

Below is the example code snippet using frame name.

public void switchToFrame(String frame) {  
 try {  
 driver.switchTo().frame(frame);  
 System.out.println("Navigated to frame with name " + frame);  
 } catch (NoSuchFrameException e) {  
 System.out.println("Unable to locate frame with id " + frame  
 + e.getStackTrace());  
 } catch (Exception e) {  
 System.out.println("Unable to navigate to frame with id " + frame  
 + e.getStackTrace());  
 }  
 }

**driver.switchTo().frame(WebElement frameElement);**

Select a frame using its previously located WebElement.

*Parameters: frameElement - The frame element to switch to.*

*Returns: driver focused on the given frame (current frame).*

*Throws: NoSuchFrameException - If the given element is neither an iframe nor a frame element. AndStaleElementReferenceException - If the WebElement has gone stale.*

Below is the example code to send an Element to the and switch.

public void switchToFrame(WebElement frameElement) {  
 try {  
 if (isElementPresent(frameElement)) {  
 driver.switchTo().frame(frameElement);  
 System.out.println("Navigated to frame with element "+ frameElement);  
 } else {  
 System.out.println("Unable to navigate to frame with element "+ frameElement);  
 }  
 } catch (NoSuchFrameException e) {  
 System.out.println("Unable to locate frame with element " + frameElement + e.getStackTrace());  
 } catch (StaleElementReferenceException e) {  
 System.out.println("Element with " + frameElement + "is not attached to the page document" + e.getStackTrace());  
 } catch (Exception e) {  
 System.out.println("Unable to navigate to frame with element " + frameElement + e.getStackTrace());  
 }  
 }

Sometimes when there are multiple Frames (Frame in side a frame), we need to first switch to the parent frame and then we need to switch to the child frame. below is the code snippet to work with multiple frames.

public void switchToFrame(String ParentFrame, String ChildFrame) {  
 try {  
 driver.switchTo().frame(ParentFrame).switchTo().frame(ChildFrame);  
 System.out.println("Navigated to innerframe with id " + ChildFrame  
 + "which is present on frame with id" + ParentFrame);  
 } catch (NoSuchFrameException e) {  
 System.out.println("Unable to locate frame with id " + ParentFrame  
 + " or " + ChildFrame + e.getStackTrace());  
 } catch (Exception e) {  
 System.out.println("Unable to navigate to innerframe with id "  
 + ChildFrame + "which is present on frame with id"  
 + ParentFrame + e.getStackTrace());  
 }  
 }

After working with the frames, main important is to come back to the web page. if we don't switch back to the default page, driver will throw an exception. Below is the code snippet to switch back to the default content.

public void switchtoDefaultFrame() {  
 try {  
 driver.switchTo().defaultContent();  
 System.out.println("Navigated back to webpage from frame");  
 } catch (Exception e) {  
 System.out  
 .println("unable to navigate back to main webpage from frame"  
 + e.getStackTrace());  
 }  
 }

Please find the selenium google code Url [Webdriver target](https://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html) for more information.

#### [2.2.25.2 How to handle javascript alerts, confirmation and prompts?](http://seleniumeasy.com/selenium-tutorials/how-to-handle-javascript-alerts-confirmation-prompts)

Generally JavaScript popups are generated by web application and hence they can be easily controlled by the browser.

Webdriver offers the ability to cope with javascript alerts using Alerts API[Click here to view Alert API Details](https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/Alert.html)

// Get a handle to the open alert, prompt or confirmation

**Alert alert = driver.switchTo().alert();**

Alert is an interface. There below are the methods that are used

//Will Click on OK button.

**alert.accept();**

// Will click on Cancel button.

**alert.dismiss()**

//will get the text which is present on th Alert.

**alert.getText();**

//Will pass the text to the prompt popup

**alert.sendkeys();**

//Is used to Authenticate by passing the credentials

alert.authenticateUsing(Credentials credentials)

## 2.3 Working with Alerts using Selenium Webdriver:

The below is the sample code for alerts, please copy and make an html file and pass it to the webdriver.

<**html**>  
<**head**>  
<**title**>Selenium Easy Alerts Sample </**title**>  
</**head**>  
<**body**>  
<**h2**>Alert Box Example</**h2**>  
<**fieldset**>  
<**legend**>Alert Box</**legend**><**p**>Click the button to display an alert box.</**p**>  
<**button** onclick="alertFunction()">Click on me</**button**>  
<**script**>  
**function** **alertFunction**()  
{  
alert("I am an example for alert box!");  
}  
</**script**>  
</**fieldset**>  
</**body**>  
</**html**>

The below program will demonstrate you working on Alerts popup using above html file.

**import** org.openqa.selenium.Alert;  
**import** org.openqa.selenium.By;  
**import** org.openqa.selenium.WebDriver;  
**import** org.openqa.selenium.firefox.FirefoxDriver;  
**import** org.testng.annotations.Test;  
  
**public** **class** **PopupsHandling** {  
 WebDriver driver=**new** FirefoxDriver();  
 @Test  
 **public** **void** **ExampleForAlert**() **throws** InterruptedException  
 {  
 driver.manage().window().maximize();  
 driver.get("file:///C:/path/alerts.html");  
 Thread.sleep(2000);  
 driver.findElement(By.xpath("//button[@onclick='alertFunction()']")).click();  
 Alert alert=driver.switchTo().alert();  
 System.out.println(alert.getText());  
 alert.accept();  
 }  
}

## 2.4 Working with Confirmation Popups

The below is the sample code for confirmation Popup, please copy and make an html file and pass it to the webdriver as below.

<**html**>  
<**head**>  
<**title**>Selenium Easy Confirm popup Sample </**title**>  
</**head**>  
<**body**>  
<**h2**>Confirm Box Example</**h2**>  
<**fieldset**>  
<**legend**>Confirm Box</**legend**>  
<**p**>Click the button to display a confirm box.</**p**>  
<**button** onclick="confirmFunction()">Click on me</**button**>  
<**p** id="confirmdemo"></**p**>  
<**script**>  
**function** **confirmFunction**()  
{  
**var** cb;  
**var** c=confirm("I am an Example for Confirm Box.\n Press any button!");  
**if** (c==true)  
 {  
 cb="You Clicked on OK!";  
 }  
**else**  
 {  
 cb="You Clicked on Cancel!";  
 }  
document.getElementById("confirmdemo").innerHTML=cb;  
}  
</**script**>  
</**fieldset**>  
</**body**>  
</**html**>

The below program will demonstrate you working on Confirmation popup using above html file.

**import** org.openqa.selenium.Alert;  
**import** org.openqa.selenium.By;  
**import** org.openqa.selenium.WebDriver;  
**import** org.openqa.selenium.firefox.FirefoxDriver;  
**import** org.testng.annotations.Test;  
  
**public** **class** **PopupsHandling** {  
 WebDriver driver=**new** FirefoxDriver();  
 @Test  
 **public** **void** **ExampleForConfirmBox**() **throws** InterruptedException  
 {  
 driver.manage().window().maximize();  
 driver.get("file:///C:/path/confirmation.html");  
 Thread.sleep(2000);  
 driver.findElement(By.xpath("//button[@onclick='confirmFunction()']")).click();  
 Alert alert=driver.switchTo().alert();  
 System.out.println(alert.getText());  
 alert.dismiss();  
 }  
}

## 2.5 Working with Prompt Popups.

In prompt, you can enter the text using webdriver sendkeys("text..")

The below is the sample code for prompt popup, please copy and make an html file and pass it to the webdriver as below.

<**html**>  
<**head**>  
<**title**>Selenium Easy Prompt popup Sample </**title**>  
</**head**>  
<**body**>  
<**h2**>Prompt Box Example</**h2**>  
<**fieldset**>  
<**legend**>Prompt Box</**legend**>  
<**p**>Click the button to demonstrate the prompt box.</**p**>  
<**button** onclick="promptFunction()">Click on me</**button**>  
<**p** id="promptdemo"></**p**>  
<**script**>  
**function** **promptFunction**()  
{  
**var** x;  
**var** person=prompt("Please enter your name","Your name");  
**if** (person!=null)  
 {  
 x="Hello " + person + "! Welcome to Selenium Easy..";  
 document.getElementById("promptdemo").innerHTML=x;  
 }  
}  
</**script**>  
</**fieldset**>  
</**body**>  
</**html**>

The below program will demonstrate you working on prompt popup using above html file.

**import** org.openqa.selenium.Alert;  
**import** org.openqa.selenium.By;  
**import** org.openqa.selenium.WebDriver;  
**import** org.openqa.selenium.firefox.FirefoxDriver;  
**import** org.testng.annotations.Test;  
  
**public** **class** **PopupsHandling** {  
 WebDriver driver=**new** FirefoxDriver();  
   
 @Test  
 **public** **void** **ExampleForPromptBox**() **throws** InterruptedException  
 {  
 driver.manage().window().maximize();  
 driver.get("file:///C:/path/prompt.html");  
 Thread.sleep(2000);  
 driver.findElement(By.xpath("//button[@onclick='promptFunction()']")).click();  
 Alert alert=driver.switchTo().alert();  
 driver.switchTo().alert().sendKeys("Helllo");  
 alert.accept();  
 System.out.println(alert.getText());  
 }  
}

### [2.5.1 Handle windows popups using Selenium Webdriver](http://seleniumeasy.com/selenium-tutorials/how-to-handle-windows-popups-using-selenium-webdriver)

There are many cases, where a application displays multiple windows when you open a website. Those are may be advertisements or may be a kind of information showing on popup windows. We can handle multiple windows using Windows Handlers in selenium webdriver.

**Step 1:** After opening the website, we need to get the main window handle by using driver.getWindowHandle();

The window handle will be in a form of lengthy alphanumeric

**Step 2:** We now need to get all the window handles by using driver.getWindowHandles();

**Step 3:** We will compare all the window handles with the main Window handles and perform the operation the window which we need.

Click here to view [Performing operations on multiple windows](http://seleniumeasy.com/selenium-tutorials/perform-operations-on-new-window-using-webdriver) using reusable methods.

The below example shows how to handle multiple windows and close all the child windows which are not need. We need to compare the main window handle to all the other window handles and close them.

**package** com.pack;  
  
**import** java.util.Set;  
**import** org.openqa.selenium.WebDriver;  
**import** org.openqa.selenium.firefox.FirefoxDriver;  
**import** org.testng.Assert;  
**import** org.testng.annotations.Test;  
  
**public** **class** **WindowExamples** {  
 **static** WebDriver driver;  
  
 @Test  
 **public** **void** **test\_CloseAllWindowsExceptMainWindow**() {  
 driver = **new** FirefoxDriver();  
 *// It will open Naukri website with multiple windows*  
 driver.get("[http://www.naukri.com/"](http://www.naukri.com/));  
   
 *// To get the main window handle*  
 String windowTitle= getCurrentWindowTitle();  
 String mainWindow = getMainWindowHandle(driver);  
 Assert.assertTrue(closeAllOtherWindows(mainWindow));  
 Assert.assertTrue(windowTitle.contains("Jobs - Recruitment"), "Main window title is not matching");  
 }  
   
 **public** String **getMainWindowHandle**(WebDriver driver) {  
 **return** driver.getWindowHandle();  
 }  
  
 **public** String **getCurrentWindowTitle**() {  
 String windowTitle = driver.getTitle();  
 **return** windowTitle;  
 }  
   
 *//To close all the other windows except the main window.*  
 **public** **static** **boolean** **closeAllOtherWindows**(String openWindowHandle) {  
 Set<String> allWindowHandles = driver.getWindowHandles();  
 **for** (String currentWindowHandle : allWindowHandles) {  
 **if** (!currentWindowHandle.equals(openWindowHandle)) {  
 driver.switchTo().window(currentWindowHandle);  
 driver.close();  
 }  
 }  
   
 driver.switchTo().window(openWindowHandle);  
 **if** (driver.getWindowHandles().size() == 1)  
 **return** **true**;  
 **else**  
 **return** **false**;  
 }  
}

### [2.5.2 Handling Authentication Window with WebDriver](http://seleniumeasy.com/selenium-tutorials/how-to-handle-authentication-popup-in-selenium-webdriver)

When you are working in a test environment, Stage or Pre Production, there are cases where you may need to work with applications which are secured with Authentication (Basic Auth).

When ever you enter the URL, it will prompt you to enter the User name and the password and It will not allow to perform any further operations until you provide username and password. And this Authentication pop-up is not a JavaScript pop-up, it is a Browser dialog window which selenium cannot handle simply using sendKeys method which we do for normal JavaScript pop-ups..

To work with Basic Authentication pop-up (which is a browser dialogue window), you just need to send the user name and password along with the application URL.

**Syntax:**

driver.**get**("[http://admin:admin@yoururl.com"](about:blank));

### 2.5.3 Check out the example below to execute in Firefox browser:

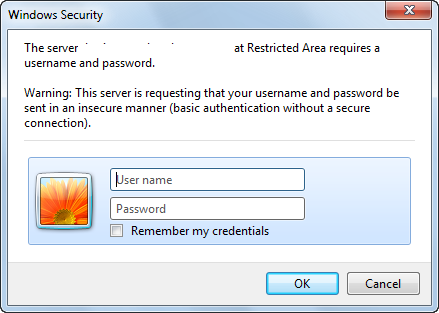
@Test  
 public void testBasicAuth\_Firefox() {  
  
 WebDriver driver = new FirefoxDriver();  
 driver.manage().window().maximize();  
 driver.get("[http://admin:admin@yoururl.com"](about:blank));  
 //To check if we have landed in the correct place  
 String text = driver.findElement(By.className("home")).getText();  
 Assert.assertTrue(text.contains("Welcome"), "Basic Authentication failed");  
  
 }

### 2.5.4 Check out the example below to work with Chrome browser:

@Test  
 public void testBasicAuth\_Chrome() {  
  
 System.setProperty("webdriver.chrome.driver", "G:/Jars/chromedriver.exe");  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.get("[http://admin:admin@yoururl.com"](about:blank));  
 //To check if we have landed in the correct place  
 String text = driver.findElement(By.className("home")).getText();  
 Assert.assertTrue(text.contains("Welcome"), "Basic Authentication failed");  
 }

The best way to make it work with IE is using AutoIt tool. If not, you may need to change the stuff in registry, To change for the current user, you need to edit in 'HKEY\_CURRENT\_USER...' and if you want to do that for all users, you can set the value of register keys as 'HKEY\_LOCAL\_MACHINE...' etc.

Once you open the URL in IE it will look like the below screen shot: -



### 2.5.5 Check out the example to work with IE using AutoIt tool.

First create AutoIt script as below and save it as basicauth.au3

; To **pass** user name **and** password  
WinWaitActive("Windows Security")  
Send("admin")  
Send("{TAB}")  
Send("admin")  
Send("{ENTER}")

After creating the above AutoIT script, compile the script and take the location of the script exe file. Now the selenium code should look like below :

@Test  
 public void testBasicAuth\_IE() {  
 DesiredCapabilities caps = DesiredCapabilities.internetExplorer();   
 caps.setCapability(InternetExplorerDriver.INITIAL\_BROWSER\_URL, "");  
  
 System.setProperty("webdriver.ie.driver", "G:/Jars/IEDriverServer.exe");  
 WebDriver driver=new InternetExplorerDriver(caps);   
 driver.manage().window().maximize();  
 driver.get("[http://yoururl.com"](http://yoururl.com/));  
 try {  
 Runtime.getRuntime().exec("G:/basicauth.exe");  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }

#### [2.5.5.1 Perform operations on new window](http://seleniumeasy.com/selenium-tutorials/perform-operations-on-new-window-using-webdriver)

There are cases where we need to open new window and perform operations or there may be cases where after clicking on any button / link, it opens new window and need to perform operations on the new window.

Let us look into such example:

Test case: We need to open '[http://linkedin.com](http://linkedin.com/)' and click on 'Help Center' link at the bottom which will open new window.

1. Verify the title of the new window

2. Verify text 'Welcome' on the page.

3. Search for a Question with text "Frequently Asked Questions" and verify the result.

package com.pack;  
  
import java.util.Set;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.WebDriverWait;  
import org.testng.Assert;  
import org.testng.annotations.Test;  
  
public class WindowExamples {  
 public static WebDriver driver;  
  
 @Test  
 public void verifySearchInNewWindow() throws InterruptedException {  
 driver = new FirefoxDriver();  
 driver.navigate().to("[http://linkedin.com/"](http://linkedin.com/));  
 driver.manage().window().maximize();  
 String mainHandle = driver.getWindowHandle();  
   
 //Wait for the element to be present  
 WebDriverWait wait = new WebDriverWait(driver, 5);  
 wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".cust-svc-link")));  
 driver.findElement(By.linkText("Help Center")).click();  
   
 //Switch to new window and verify the title  
 waitForNewWindowAndSwitchToIt(driver);  
 String newTitle = getCurrentWindowTitle();  
 Assert.assertEquals(newTitle, "LinkedIn Help Center", "New window title is not matching");  
   
 //Verify the text present on the page  
 String textOnpage=driver.findElement(By.cssSelector(".welcome")).getText().trim();  
 Assert.assertEquals(textOnpage, "Welcome!");  
   
 //Verify search text on the page  
 String searchText="Frequently Asked Questions";  
 WebElement searchInputBox=driver.findElement(By.id("kw"));  
 searchInputBox.sendKeys(searchText);  
   
 WebElement searchButton = driver.findElement(By.cssSelector(".button.leftnoround.blue"));  
 searchButton.click();  
   
 WebElement resultedElement = driver.findElement(By.cssSelector(".rn\_Element2"));  
 String resultedText = resultedElement.getText().trim();  
 System.out.println(resultedText);  
 Assert.assertTrue(resultedText.contains(searchText), "Search successfull");  
   
 closeAllOtherWindows(driver, mainHandle);  
 }

To execute the above test we have created a methods which can be reused with multiple tests.

**Below method is used to get the main window handle. we will the driver as parameter.**

**public** **static** String **getMainWindowHandle**(WebDriver driver) {  
 **return** driver.getWindowHandle();  
 }

**Below method is used to get the current window title**

**public** **static** String **getCurrentWindowTitle**() {  
 String windowTitle = driver.getTitle();  
 **return** windowTitle;  
 }

**Below method is used to close all the other windows except the main window.**

public static boolean closeAllOtherWindows(WebDriver driver, String openWindowHandle) {  
 Set<String> allWindowHandles = driver.getWindowHandles();  
 for (String currentWindowHandle : allWindowHandles) {  
 if (!currentWindowHandle.equals(openWindowHandle)) {  
 driver.switchTo().window(currentWindowHandle);  
 driver.close();  
 }  
 }  
   
 driver.switchTo().window(openWindowHandle);  
 if (driver.getWindowHandles().size() == 1)  
 return true;  
 else  
 return false;  
 }

**Below method is used to wait for the new window to be present and switch to it.**

public static void waitForNewWindowAndSwitchToIt(WebDriver driver) throws InterruptedException {  
 String cHandle = driver.getWindowHandle();  
 String newWindowHandle = null;  
 Set<String> allWindowHandles = driver.getWindowHandles();  
   
 //Wait for 20 seconds for the new window and throw exception if not found  
 for (int i = 0; i < 20; i++) {  
 if (allWindowHandles.size() > 1) {  
 for (String allHandlers : allWindowHandles) {  
 if (!allHandlers.equals(cHandle))  
 newWindowHandle = allHandlers;  
 }  
 driver.switchTo().window(newWindowHandle);  
 break;  
 } else {  
 Thread.sleep(1000);  
 }  
 }  
 if (cHandle == newWindowHandle) {  
 throw new RuntimeException(  
 "Time out - No window found");  
 }  
 }  
}

Note: When ever we work on multiple windows, switching plays major role. We should switch to the desired window to perform operations and again switch back to default window to work on main window.

### 2.5.6 Upload and Download Examples:

#### [2.5.6.1 Uploading a file with sendKeys method](http://seleniumeasy.com/selenium-tutorials/uploading-file-with-sendkeys-method)

The Easy way of uploading a file is simple case of just finding the element and typing the absolute path of the document into it.

It is mandatory that it works only when the element has <input type=”file”> is enabled. So please make sure that the input element is visible. In the below example 'uploadsubmit' is the name of that element and in '**sendKeys()**' we have to specify the absolute path of the content that we want to upload (that can be file/image/video etc).

Sample HTML Code should look similar to this :

<**html**>  
<**body**>  
<**form** enctype="multipart/form-data" action="parse\_file.php" method="post">   
 <**p**>Browse for a file to upload: </**p**>  
 <**input** type="file" name="uploadsubmit">  
 <**br**/><**br**/>  
 <**input** type="submit" value="SUBMIT">  
</**form**>  
</**body**>  
</**html**>

The following is the example program to upload a file using sendKeys() in selenium webdriver without using any third party tools:

**package** com.easy.upload;  
**import** org.openqa.selenium.By;  
**import** org.openqa.selenium.WebDriver;  
**import** org.openqa.selenium.WebElement;  
**import** org.openqa.selenium.firefox.FirefoxDriver;  
**import** org.testng.Assert;  
**import** org.testng.annotations.Test;  
  
**public** **class** **UploadFile** {  
  
 **static** WebDriver driver;  
 String URL = "application URL";  
 @Test  
 **public** **void** **testUpload**() **throws** InterruptedException  
 {  
 driver = **new** FirefoxDriver();  
 driver.get(URL);  
 WebElement element = driver.findElement(By.name("uploadsubmit"));  
*//To input the filename along with path*  
 element.sendKeys("D:/file.txt");  
*// To click on the submit button (Not the browse button)*  
 driver.findElement(By.name("SubmitBtn")).click();  
 String checkText = driver.findElement(By.id("message")).getText();  
 Assert.assertEquals("File uploaded successfully", checkText);  
 }   
}

Make sure, you are not clicking on the browse button, clicking on browse button will open windows dialogue box where selenium webdriver will won't work. Using above code we have successfully given the file path. Now submit the form/upload and see that file has successfully uploaded.

#### [2.5.6.2 Upload file using AutoIT](http://seleniumeasy.com/selenium-tutorials/upload-a-file-using-selenium-webdriver-with-autoit)

#### [2.5.6.3 File upload using Robots](http://seleniumeasy.com/selenium-tutorials/webdriver-file-upload-using-robots)

We have discussed uploading a file using [using Webdriver Sendkeys method](http://seleniumeasy.com/selenium-tutorials/uploading-file-with-sendkeys-method) and [Using AutoIT Tool](http://seleniumeasy.com/selenium-tutorials/upload-a-file-using-selenium-webdriver-with-autoit). Now here we will look into an other way of doing file upload using [Robot class](http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html) and [StringSelection class](https://docs.oracle.com/javase/7/docs/api/java/awt/datatransfer/StringSelection.htm) in java.

Robot class is used to (generate native system input events) take the control of mouse and keyboard. Once you get the control, you can do any type of operation related to mouse and keyboard through with java code.

There are different methods which robot class uses. Here in the below example we have used *'keyPress'* and *'keyRelease'* methods.

*keyPress* - takes keyCode as Parameter and Presses here a given key.

*keyrelease* - takes keyCode as Parameterand Releases a given key

Both the above methods *Throws - IllegalArgumentException, if keycode is not a valid key.*

We have defined two methods in the below example along with the test which is used to upload a file.

**package** com.easy.upload;  
  
**import** java.awt.Robot;  
**import** java.awt.Toolkit;  
**import** java.awt.datatransfer.StringSelection;  
**import** java.awt.event.KeyEvent;  
  
**import** org.openqa.selenium.By;  
**import** org.openqa.selenium.WebDriver;  
**import** org.openqa.selenium.WebElement;  
**import** org.openqa.selenium.firefox.FirefoxDriver;  
**import** org.testng.annotations.Test;  
  
**public** **class** **UploadFileRobot** {  
  
 String URL = "application URL";  
 @Test  
 **public** **void** **testUpload**() **throws** InterruptedException  
 {  
 WebDriver driver = **new** FirefoxDriver();  
 driver.get(URL);  
 WebElement element = driver.findElement(By.name("uploadfile"));  
 element.click();  
 uploadFile("path to the file");  
 Thread.sleep(2000);  
 }  
   
 /\*\*  
 \* This method will set any parameter string to the system's clipboard.  
 \*/  
 **public** **static** **void** **setClipboardData**(String string) {  
 *//StringSelection is a class that can be used for copy and paste operations.*  
 StringSelection stringSelection = **new** StringSelection(string);  
 Toolkit.getDefaultToolkit().getSystemClipboard().setContents(stringSelection, **null**);  
 }  
   
 **public** **static** **void** **uploadFile**(String fileLocation) {  
 **try** {  
 *//Setting clipboard with file location*  
 setClipboardData(fileLocation);  
 *//native key strokes for CTRL, V and ENTER keys*  
 Robot robot = **new** Robot();  
   
 robot.keyPress(KeyEvent.VK\_CONTROL);  
 robot.keyPress(KeyEvent.VK\_V);  
 robot.keyRelease(KeyEvent.VK\_V);  
 robot.keyRelease(KeyEvent.VK\_CONTROL);  
 robot.keyPress(KeyEvent.VK\_ENTER);  
 robot.keyRelease(KeyEvent.VK\_ENTER);  
 } **catch** (Exception exp) {  
 exp.printStackTrace();  
 }  
 }  
}

#### [2.5.6.4 Download file using selenium webdriver](http://seleniumeasy.com/selenium-tutorials/how-to-download-a-file-with-webdriver)

### 

### 2.5.7 Mouse Hover Actions in Selenium Webdriver

*//hover over Categories. Hover in multiple windows.*

Actions act = **new** Actions(*driver*);

act.moveToElement(*categories*).moveToElement(**machinery**).build().perform();

**handTools**.click();

http://seleniumeasy.com/selenium-tutorials/how-to-perform-mouseover-action-in-selenium-webdriver

### 2.5.8 Set browser width and height in Selenium Webdriver

System.out.println(driver.manage().window().getSize());  
 Dimension d = **new** Dimension(420,600);  
 *//Resize the current window to the given dimension*  
 driver.manage().window().setSize(d);

[Set browser width and height in Selenium Webdriver](http://seleniumeasy.com/selenium-tutorials/set-browser-width-and-height-in-selenium-webdriver)

### 2.5.9 How to minimize the browser?

driver.manage().window().setPosition(new Point(-2000, 0))

### 2.5.10 Resizing a web element using movebyoffset

Normally when ever we want to change the size of an element we do with the help of mouse manually. Now we will see to resize / change the size of an element using webdriver actions class with moveByOffset which moves the mouse position from its current position by the given offset.

Below are the simple steps thats needs to be followed for the below example:-

Step 1: Open the URL

Step 2: Wait for the element that you want to resize. (Make sure if there are any frames then we need to shift to the frame and then perform operation).

Step 3: We will define a method using which we need to pass web element and the coordinates to the method.

We will now perform the below example on [Jquery Website](http://jqueryui.com/resizable/) which has Resizable element.

package com.pack.jquery;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.NoSuchElementException;  
import org.openqa.selenium.StaleElementReferenceException;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.interactions.Actions;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.WebDriverWait;  
import org.testng.annotations.Test;  
  
public class ResizeExample {  
 WebDriver driver;  
  
 @Test  
 public void testToResizeElement() {  
  
 driver = new FirefoxDriver();  
 driver.manage().window().maximize();  
 driver.navigate().to("[http://jqueryui.com/resizable/"](http://jqueryui.com/resizable/));  
 WebDriverWait wait = new WebDriverWait(driver, 5);  
 wait.until(ExpectedConditions.frameToBeAvailableAndSwitchToIt(By.cssSelector(".demo-frame")));  
 WebElement resizeableElement = driver.findElement(By.cssSelector(".ui-resizable-handle.ui-resizable-se.ui-icon.ui-icon-gripsmall-diagonal-se"));  
 resize(resizeableElement, 50, 50);  
 }  
  
 public void resize(WebElement elementToResize, int xOffset, int yOffset) {  
 try {  
 if (elementToResize.isDisplayed()) {  
 Actions action = new Actions(driver);  
 action.clickAndHold(elementToResize).moveByOffset(xOffset, yOffset).release().build().perform();  
 } else {  
 System.out.println("Element was not displayed to drag");  
 }  
 } catch (StaleElementReferenceException e) {  
 System.out.println("Element with " + elementToResize + "is not attached to the page document " + e.getStackTrace());  
 } catch (NoSuchElementException e) {  
 System.out.println("Element " + elementToResize + " was not found in DOM " + e.getStackTrace());  
 } catch (Exception e) {  
 System.out.println("Unable to resize" + elementToResize + " - " + e.getStackTrace());  
 }  
 }  
  
}

<http://www.seleniumeasy.com/selenium-tutorials/resizing-a-web-element-using-webdriver-action-class-movebyoffset>

### 2.5.11 Handling SSL certificates in Firefox and Chrome browser

#### 2.5.11.1 WHAT IS SSL CERTIFICATE ?

*SSL is used to keep sensitive information which is sent across the Internet encrypted so that only the intended recipient understand it.* This is important because, the information that we send on the internet is passed from one system to other system to the destination server.

If it is not encrypted with an SSL certificate, any computer in between you and the destination server can see your private information such as credit card numbers, usernames, passwords and other sensitive information.

When an SSL certificate is used, the information will be encrypted and it will ensure that you are sending information to the right server and not to a criminal’s server.

#### 2.5.11.2 WHEN DO WE GET UNTRUSTED CONNECTION ERROR

When ever you try to visit a website whose web address starts with https, your communication with this site is encrypted to ensure your privacy. Before starting the encrypted communication the website, you will be presented with a "certificate" to identify itself.

The certificate helps to determine whether the site you are visiting is actually the site that it claims to be. If there is any problem with the certificate, you will see an alert saying 'This Connection Is Untrusted'. What it means is that browser (Firefox/Chrome etc) isn't able to verify the identity of the website, there can be several problems which can cause browser to reject a certificate.

The below are the common errors that we see :

#### 2.5.11.3 Certificate will not be valid until (date)

Error code: sec\_error\_expired\_issuer\_certificate

This error can occur if our system clock has the wrong date, check error message which will be in the past. We can fix this problem, by setting system clock to current date.

#### 2.5.11.4 Certificate expired on (date)

Error code: sec\_error\_expired\_certificate

This error occurs when a website identity certification has expired. This can also occur if system clock has the wrong date. We can fix this problem, by setting system clock to current date.

#### 2.5.11.5 Certificate is only valid for (site name)

Error code: ssl\_error\_bad\_cert\_domain

This error says that the identification sent to you by the site is actually for another site. While anything you send would be safe from eavesdroppers , the recipient may not be the same who you think it is.

The above listed errors are the common errors, you may come across other errors which actually depends on the websites that you access.

We will create new firefox profile and set 'setAcceptUntrustedCertificates' as true and setAssumeUntrustedCertificateIssuer as false.

Firefox:

|  |
| --- |
| FirefoxProfile profile = new FirefoxProfile();  profile.setAcceptUntrustedCertificates(true);   profile.setAssumeUntrustedCertificateIssuer(false);  driver = new FirefoxDriver(profile);   driver.manage().window().maximize(); |

Chrome Browser:

|  |
| --- |
| DesiredCapabilities capability = DesiredCapabilities.chrome();  // To Accept SSL certificate  capability.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS, true);  // setting system property for Chrome browser  System.setProperty("webdriver.chrome.driver", "E:/chromedriver.exe");  // create Google Chrome instance and maximize it  driver = new ChromeDriver(capability);  driver.manage().window().maximize(); |

### 2.5.12 How to add extensions as desired capabilities in Chrome Browser?

### 2.5.13 Capabilities & ChromeOptions

|  |
| --- |
| // Add extention  ChromeOptions options = new ChromeOptions(); options.addExtensions(new File("/path/to/extension.crx")); ChromeDriver driver = new ChromeDriver(options);  //Add the WebDriver proxy capability.  ChromeOptions options = **new** ChromeOptions();  // Add the WebDriver proxy capability. Proxy proxy = **new** Proxy(); proxy.setHttpProxy("myhttpproxy:3337"); options.setCapability("proxy", proxy);  // Add a ChromeDriver-specific capability. options.addExtensions(**new** File("/path/to/extension.crx")); ChromeDriver driver = **new** ChromeDriver(options); |

<https://sites.google.com/a/chromium.org/chromedriver/capabilities>

### 

### 2.5.14 How to set capabilities of FireFox browser?

### [2.5.15 Defining Firefox Profile preferences using Selenium Webdriver](http://seleniumeasy.com/selenium-tutorials/firefox-profile-preferences-using-selenium-webdriver)

|  |
| --- |
| [FirefoxProfile profile = new FirefoxProfile(); profile.setPreference("browser.startup.homepage", "http://www.google.com");](http://seleniumeasy.com/selenium-tutorials/firefox-profile-preferences-using-selenium-webdriver)  HYPERLINK "http://seleniumeasy.com/selenium-tutorials/firefox-profile-preferences-using-selenium-webdriver" |

<https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/firefox/FirefoxProfile.html>

### 2.5.16 How To Handle AJAX Calls Using Selenium WebDriver

## 2.6 Handle Ajax Calls using Selenium:

Handling AJAX calls is one of the common issues when using Selenium WebDriver. We wouldn’t know when the AJAX call would get completed and the page has been updated. In this post, we see how to handle AJAX calls using Selenium.

AJAX stands for Asynchronous JavaScript and XML. AJAX allows the web page to retrieve small amounts of data from the server without reloading the entire page. AJAX sends HTTP requests from the client to server and then process the server’s response without reloading the entire page. To handle AJAX controls, wait commands may not work. It’s just because the actual page is not going to refresh.

When you click on a submit button, required information may appear on the web page without refreshing the browser. Sometimes it may load in a second and sometimes it may take longer. **We have no control on loading time. The best approach to handle this kind of situations in selenium is to use dynamic waits** (i.e. WebDriverWait in combination with ExpectedCondition)

Some of the methods which are available are as follows:

Check this post for detailed explanation on [WebDeriverWait](https://www.softwaretestingmaterial.com/webdriverwait-selenium-webdriver/)

In the above post, we have discussed on titleIs() condition. Let’s see some other conditions in detail.

|  |
| --- |
| wait.until(ExpectedConditions.titleIs("Deal of the Day")); wait.until(ExpectedConditions.elementToBeClickable(By.xpath("xpath"))); wait.until(ExpectedConditions.alertIsPresent()) !=null); |

### 2.6.1 StaleElementReference Exceptions in Selenium Webdriver

This Exception occurs when driver is trying to perform action on the element which is no longer exists or not valid.

WebElement ele = driver.findElement(By.id("sample"));

// Before clicking some thing happened and DOM has changed due to page refresh, or element is removed and re-added

ele.click();

Now at this point, the element which you're clicking is no longer valid.

so it throws up its hands and gives control to user, who as the test/app author should know exactly what may or may not happen.

In order overcome this, we need to explicitly wait until the DOM is in a state where we are sure that DOM won't change.

When Javascript / Ajax updates the page between the findElement and the click call then will get a StaleElementException. Here the reference to the element in the DOM that previously had becomes stale and we can no longer able to use this reference (click call) to interact with the element in the DOM.

Try to get around this by first using an explicit wait on the element to ensure the ajax call is complete, then get a reference to the element again.

[StaleElementReferenceExceptions in Selenium Webdriver](http://seleniumeasy.com/selenium-tutorials/staleelementreferenceexception-in-selenium-webdriver)

### 2.6.2 Handale Cookies

Selenium webdriver can perform required task with respect to browser cookies. We add , delete, delete particular cookie by passing the name, and so on. Let us look all of the in details.

First let us see how add cookie works:

**Add Cookie**

Method Name: addCookie(Cookie cookie)

Syntax:driver.**manage().addCookie(arg0);**

Purpose: To add a specific cookie into cookies. If the cookie's domain name is left blank, it is assumed that the cookie is meant for the domain of the current document.

Parameters: cookie - The name of the cookie to add.

[Click here for example program](http://seleniumeasy.com/selenium-tutorials/how-to-add-cookie-with-selenium-webdriver)

**Delete Cookie**

Method Name: deleteCookie(Cookie cookie)

Syntax: **driver.manage().deleteCookie(arg0);**

Purpose: Delete a cookie from the browser's "cookie jar". The domain of the cookie will be ignored.

Parameter: Cookie

[Click here for example program](http://seleniumeasy.com/selenium-tutorials/how-to-delete-cookies-in-selenium-webdriver)

**Delete Cookie with Name**

Method Name: deleteCookieNamed(java.lang.String name)

Syntax: **driver.manage().deleteCookieNamed(arg0);**

Purpose: Delete the named cookie from the current domain. This is equivalent to setting the named cookie's expiry date to sometime in the past.

Parameters: name - The name of the cookie to delete

[Click here for example program](http://seleniumeasy.com/selenium-tutorials/how-to-delete-cookies-in-selenium-webdriver)

**Delete All Cookies**

Method Name: deleteAllCookies()

Syntax: **driver.manage().deleteAllCookies();**

Purpose: It will delete all the cookies for the current domain.

Parameters: N/A

[Click here for example program](http://seleniumeasy.com/selenium-tutorials/how-to-delete-cookies-in-selenium-webdriver)

**Get Cookies**

Method Name: getCookies()

Syntax: **driver.manage().getCookies();**

Purpose: Get all the cookies for the current domain. This is the equivalent of calling "document.cookie" and parsing the result.

Returns: A Set of cookies for the current domain.

**Get the Cookie with Specific Name**

Method Name: getCookieNamed(java.lang.String name)

Syntax: **driver.manage().getCookieNamed(arg0);**

Purpose: To Get a cookie with a given name.

Parameters: name - the name of the cookie

Returns: It will return the cookie value for the name specified, or null if no cookie found with the given name

### 2.6.3 Taking ScreenShot ONLY for Failed Tests

### 

|  |
| --- |
| package com.pack.listeners;  import java.io.File; import java.io.IOException;  import org.apache.commons.io.FileUtils; import org.openqa.selenium.OutputType; import org.openqa.selenium.TakesScreenshot; import org.openqa.selenium.WebDriver; import org.testng.ITestContext; import org.testng.ITestListener; import org.testng.ITestResult;  import com.pack.sample.TestBase;  public class TestListener implements ITestListener {  WebDriver driver=null;  String filePath = "D:\\SCREENSHOTS";  @Override  public void onTestFailure(ITestResult result) {  System.out.println("\*\*\*\*\* Error "+result.getName()+" test has failed \*\*\*\*\*");  String methodName=result.getName().toString().trim();  takeScreenShot(methodName);  }    public void takeScreenShot(String methodName) {  *//get the driver*  driver=TestBase.getDriver();  File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  *//The below method will save the screen shot in d drive with test method name*   try {  FileUtils.copyFile(scrFile, new File(filePath+methodName+".png"));  System.out.println("\*\*\*Placed screen shot in "+filePath+" \*\*\*");  } catch (IOException e) {  e.printStackTrace();  }  }  public void onFinish(ITestContext context) {}    public void onTestStart(ITestResult result) { }    public void onTestSuccess(ITestResult result) { }   public void onTestSkipped(ITestResult result) { }   public void onTestFailedButWithinSuccessPercentage(ITestResult result) { }   public void onStart(ITestContext context) { } } |

Before executing the above program, we need to add *TestListener* class in testng.xml file as below:

|  |
| --- |
| <listeners>  <listener class-name="com.pack.listeners.TestListener"/>  </listeners> |

[Handling keyboard events and mouse hover events using Webdriver](http://seleniumeasy.com/selenium-tutorials/handling-keyboard-events-and-mouse-hover-events-using-webdriver)

[Uploading a file with Selenium Webdriver](http://seleniumeasy.com/selenium-tutorials/uploading-file-with-selenium-webdriver)

[Read data from Properties file using Java Selenium](http://seleniumeasy.com/selenium-tutorials/read-data-from-properties-file-using-java-selenium)

### 2.6.4 How to get x,y coordinates of an elements?

|  |
| --- |
| package softwareTestingMaterial; import org.openqa.selenium.By; import org.openqa.selenium.Point; import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement; import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.interactions.Actions; public class TestTestTest {    public static void main (String [] args) throws InterruptedException {  *//Instantiation of driver object. To launch Firefox browser*  System.setProperty("webdriver.chrome.driver", "D:\\Selenium Environment\\Drivers\\chromedriver.exe");  WebDriver driver = new ChromeDriver();  *//To open url*  driver.get("https://www.softwaretestingmaterial.com/sample-webpage-to-automate/");  *//driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);*  Thread.sleep(5000);  driver.get("https://www.softwaretestingmaterial.com/sample-webpage-to-automate/");  *// find the SoftwareTestingMaterial logo using linkText Locator*  WebElement element = driver.findElement(By.linkText("Software Testing Material"));  *//Used points class to get x and y coordinates of element.*  Point point = element.getLocation();  int xcord = point.getX();  System.out.println("Position of the webelement from left side is "+xcord +" pixels");  int ycord = point.getY();  System.out.println("Position of the webelement from top side is "+ycord +" pixels");  *// Using Actions class*  Actions action = new Actions(driver);  *//clicking on the logo based on x coordinate and y coordinate*  *//you will be redirecting to the home page of softwaretestingmaterial.com*  action.moveToElement(element, xcord, ycord).click().build().perform();  */\*((JavascriptExecutor)driver).executeScript("window.scrollTo(0,"+element.getLocation().y+")");  element.click();\*/*  */\*JavascriptExecutor executor = (JavascriptExecutor) driver;  executor.executeScript("window.scroll(" + xcord + ", " + ycord + ");");  executor.executeScript("arguments[0].click();", element);\*/*  } } |

[How to perform Drag and Drop](http://seleniumeasy.com/selenium-tutorials/drag-and-drop-using-webdriver-action-class)

|  |
| --- |
| [package com.pack.dragndrop;  import org.openqa.selenium.By; import org.openqa.selenium.NoSuchElementException; import org.openqa.selenium.StaleElementReferenceException; import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement; import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.interactions.Actions; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait; import org.testng.Assert; import org.testng.annotations.Test;  public class DragNDropExample {   WebDriver driver;    @Test  public void testDragAndDropExample() {  driver = new FirefoxDriver();  driver.manage().window().maximize();  driver.navigate().to("http://jqueryui.com/droppable/");  *//Wait for the frame to be available and switch to it*  WebDriverWait wait = new WebDriverWait(driver, 5);  wait.until(ExpectedConditions.frameToBeAvailableAndSwitchToIt(By.cssSelector(".demo-frame")));  WebElement Sourcelocator = driver.findElement(By.cssSelector(".ui-draggable"));  WebElement Destinationlocator = driver.findElement(By.cssSelector(".ui-droppable"));  dragAndDrop(Sourcelocator,Destinationlocator);  String actualText=driver.findElement(By.cssSelector("#droppable>p")).getText();  Assert.assertEquals(actualText, "Dropped!");  } }](http://seleniumeasy.com/selenium-tutorials/drag-and-drop-using-webdriver-action-class)  public void dragAndDrop(WebElement sourceElement, WebElement destinationElement) {  try {  if (sourceElement.isDisplayed() && destinationElement.isDisplayed()) {  Actions action = new Actions(driver);  action.dragAndDrop(sourceElement, destinationElement).build().perform();  } else {  System.out.println("Element was not displayed to drag");  }  } catch (StaleElementReferenceException e) {  System.out.println("Element with " + sourceElement + "or" + destinationElement + "is not attached to the page document "  + e.getStackTrace());  } catch (NoSuchElementException e) {  System.out.println("Element " + sourceElement + "or" + destinationElement + " was not found in DOM "+ e.getStackTrace());  } catch (Exception e) {  System.out.println("Error occurred while performing drag and drop operation "+ e.getStackTrace());  }  }  **public** **void** **dragAndDrop**(WebElement sourceElement, WebElement destinationElement)  {  (**new** Actions(driver)).dragAndDrop(sourceElement, destinationElement).perform();  } } HYPERLINK "http://seleniumeasy.com/selenium-tutorials/drag-and-drop-using-webdriver-action-class" |

|  |
| --- |
| In order to right click .contextClick(element) of Action class package com.pack.rightclick;  import org.openqa.selenium.Alert; import org.openqa.selenium.By; import org.openqa.selenium.NoSuchElementException; import org.openqa.selenium.StaleElementReferenceException; import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement; import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.interactions.Actions; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait; import org.testng.Assert; import org.testng.annotations.AfterClass; import org.testng.annotations.BeforeClass; import org.testng.annotations.Test;  public class RightClickExample {   WebDriver driver;    String URL = "http://medialize.github.io/jQuery-contextMenu/demo.html";    @BeforeClass  public void Setup() {  driver = new FirefoxDriver();  driver.manage().window().maximize();  }    @Test  public void rightClickTest() {  driver.navigate().to(URL);  By locator = By.cssSelector(".context-menu-one.box");  WebDriverWait wait = new WebDriverWait(driver, 5);  wait.until(ExpectedConditions.presenceOfElementLocated(locator));   WebElement element=driver.findElement(locator);  rightClick(element);  WebElement elementEdit =driver.findElement(By.cssSelector(".context-menu-item.icon.icon-edit>span"));  elementEdit.click();  Alert alert=driver.switchTo().alert();  String textEdit = alert.getText();  Assert.assertEquals(textEdit, "clicked: edit", "Failed to click on Edit link");  }    public void rightClick(WebElement element) {  try {  Actions action = new Actions(driver).contextClick(element);  action.build().perform();   System.out.println("Sucessfully Right clicked on the element");  } catch (StaleElementReferenceException e) {  System.out.println("Element is not attached to the page document "  + e.getStackTrace());  } catch (NoSuchElementException e) {  System.out.println("Element " + element + " was not found in DOM "  + e.getStackTrace());  } catch (Exception e) {  System.out.println("Element " + element + " was not clickable "  + e.getStackTrace());  }  }   @AfterClass  public void tearDown() {  driver.quit();  }   } |

### [2.6.6 Working with AutoComplete Text box](http://seleniumeasy.com/selenium-tutorials/working-with-ajax-or-jquery-auto-complete-text-box-using-webdriver)

|  |
| --- |
| [package com.pack.auto;  import java.util.List;  import org.openqa.selenium.By; import org.openqa.selenium.NoSuchElementException; import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement; import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait; import org.testng.annotations.AfterClass; import org.testng.annotations.BeforeClass; import org.testng.annotations.Test;  public class AutoCompleteExample {   WebDriver driver;  WebDriverWait wait;   String URL = "http://jqueryui.com/autocomplete/";  private By frameLocator = By.className("demo-frame");  private By tagText = By.id("tags");   @BeforeClass  public void Setup() {  driver = new FirefoxDriver();  driver.manage().window().maximize();  wait = new WebDriverWait(driver, 5);  }   @Test  public void rightClickTest() {  driver.navigate().to(URL);  WebElement frameElement=driver.findElement(frameLocator);  driver.switchTo().frame(frameElement);  wait.until(ExpectedConditions.presenceOfElementLocated(tagText));  WebElement textBoxElement = driver.findElement(tagText);  textBoxElement.sendKeys("a");  selectOptionWithText("Java");  *//selectOptionWithIndex(2);*    }](http://seleniumeasy.com/selenium-tutorials/working-with-ajax-or-jquery-auto-complete-text-box-using-webdriver)  public void selectOptionWithText(String textToSelect) {  try {  WebElement autoOptions = driver.findElement(By.id("ui-id-1"));  wait.until(ExpectedConditions.visibilityOf(autoOptions));   List<WebElement> optionsToSelect = autoOptions.findElements(By.tagName("li"));  for(WebElement option : optionsToSelect){  if(option.getText().equals(textToSelect)) {  System.out.println("Trying to select: "+textToSelect);  option.click();  break;  }  }    } catch (NoSuchElementException e) {  System.out.println(e.getStackTrace());  }  catch (Exception e) {  System.out.println(e.getStackTrace());  }  public void selectOptionWithIndex(int indexToSelect) {    try {  WebElement autoOptions = driver.findElement(By.id("ui-id-1"));  wait.until(ExpectedConditions.visibilityOf(autoOptions));   List<WebElement> optionsToSelect = autoOptions.findElements(By.tagName("li"));  if(indexToSelect<=optionsToSelect.size()) {  System.out.println("Trying to select based on index: "+indexToSelect);  optionsToSelect.get(indexToSelect).click();  }  }   catch (NoSuchElementException e) {  System.out.println(e.getStackTrace());  }  catch (Exception e) {  System.out.println(e.getStackTrace());  }  }   @AfterClass  public void tearDown() {  driver.quit();  }  } |

[Validate Downloaded file after clicking on downloaded button/ link](http://seleniumeasy.com/selenium-tutorials/verify-file-after-downloading-using-webdriver-java)

[How to verify entries in Exported CSV file](http://seleniumeasy.com/selenium-tutorials/verify-entries-in-exported-csv-file-using-webdriver-java)

[Element is not clickable at point SeleniumWebdriverException](http://seleniumeasy.com/selenium-tutorials/element-is-not-clickable-at-point-selenium-webdriver-exception)

### [2.6.7 Extract PDF text And Verify Text Present in PDF using WebDriver](http://seleniumeasy.com/selenium-tutorials/how-to-extract-pdf-text-and-verify-using-selenium-webdriver-java)

The second way is using third party library. In this example we will how to use 'Apache PDFBox' library  
  
To extract text from a PDF we can use Apache PDFBox library which is one of the main feature of PDFBox. I can extract the text from variety of PDF documents. The functionality of extracting text is encapsulated in 'org.apache.pdfbox.util.PDFTextStripper'  
  
It also provides an option to limit the text that is extracted during the extraction process by specifying the range of pages that we want to extract. For example, if the PDF has 100 pages, we can give the range from first to second page to validate the text present.

|  |
| --- |
| import java.io.BufferedInputStream; import java.io.IOException; import java.net.MalformedURLException; import java.net.URL; import junit.framework.Assert; import org.apache.pdfbox.cos.COSDocument; import org.apache.pdfbox.pdfparser.PDFParser; import org.apache.pdfbox.pdmodel.PDDocument; import org.apache.pdfbox.util.PDFTextStripper; import org.openqa.selenium.By; import org.openqa.selenium.WebDriver; import org.openqa.selenium.firefox.FirefoxDriver; import org.testng.annotations.AfterClass; import org.testng.annotations.BeforeClass; import org.testng.annotations.Test;  public class ReadPDF {    WebDriver driver;    @BeforeClass  public void setUp() {  driver = new FirefoxDriver();  }    */\*\*  \* To verify PDF content in the pdf document  \*/*  @Test  public void testVerifyPDFTextInBrowser() {    driver.get("http://www.princexml.com/samples/");  driver.findElement(By.linkText("PDF flyer")).click();  Assert.assertTrue(verifyPDFContent(driver.getCurrentUrl(), "Prince Cascading"));  }   */\*\*  \* To verify pdf in the URL   \*/*  @Test  public void testVerifyPDFInURL() {  driver.get("http://www.princexml.com/samples/");  driver.findElement(By.linkText("PDF flyer")).click();  String getURL = driver.getCurrentUrl();  Assert.assertTrue(getURL.contains(".pdf"));  }     public boolean verifyPDFContent(String strURL, String reqTextInPDF) {    boolean flag = false;    PDFTextStripper pdfStripper = null;  PDDocument pdDoc = null;  COSDocument cosDoc = null;  String parsedText = null;   try {  URL url = new URL(strURL);  BufferedInputStream file = new BufferedInputStream(url.openStream());  PDFParser parser = new PDFParser(file);    parser.parse();  cosDoc = parser.getDocument();  pdfStripper = new PDFTextStripper();  pdfStripper.setStartPage(1);  pdfStripper.setEndPage(1);    pdDoc = new PDDocument(cosDoc);  parsedText = pdfStripper.getText(pdDoc);  } catch (MalformedURLException e2) {  System.err.println("URL string could not be parsed "+e2.getMessage());  } catch (IOException e) {  System.err.println("Unable to open PDF Parser. " + e.getMessage());  try {  if (cosDoc != null)  cosDoc.close();  if (pdDoc != null)  pdDoc.close();  } catch (Exception e1) {  e.printStackTrace();  }  }    System.out.println("+++++++++++++++++");  System.out.println(parsedText);  System.out.println("+++++++++++++++++");   if(parsedText.contains(reqTextInPDF)) {  flag=true;  }    return flag;  }    @AfterClass  public void tearDown() {  driver.quit();  } } The above case works fine when the PDF file is opened in a Browser after clicking on the Print button. There are few cases where once we click on Print, it will download the pdf file. In these cases we should do in the below way: We need to change the below code URL url = new URL(strURL); BufferedInputStream file = new BufferedInputStream(url.openStream()); PDFParser parser = new PDFParser(file);   convert as below  File file = new File("D:/Paynetsbicardbill.pdf");  PDFParser parser = new PDFParser(new FileInputStream(file)); |

[How to Find Broken / Invalid Images on a Web Page](http://seleniumeasy.com/selenium-tutorials/find-broken-images-in-a-webpage-using-webdriver-java)

[Find broken links on website using webdriver and HTTP Client](http://seleniumeasy.com/selenium-tutorials/how-to-find-broken-links-using-webdriver-java)

[Verify tooltip text with selenium webdriver using java](http://seleniumeasy.com/selenium-tutorials/how-to-verify-tooltip-text-with-selenium-webdriver-using-java)

[Database testing with selenium example](http://seleniumeasy.com/selenium-tutorials/database-testing-example-with-selenium-using-java)

[Webdriver EventListener example](http://www.seleniumeasy.com/selenium-tutorials/webdriver-event-listener-example)

[How to Configure Selenium Grid](http://www.seleniumeasy.com/selenium-tutorials/how-to-configure-selenium-grid)

[Execute webdriver tests in Parallel using selenium Grid](http://www.seleniumeasy.com/selenium-tutorials/parallel-execution-in-selenium-grid)

[Launch Edge browser using Microsoft Edge Driver](https://www.seleniumeasy.com/selenium-tutorials/launching-microsoft-edge-browser-using-selenium-3)

### 2.6.8 Q: How would you select a menu item from a drop down menu?

**Ans.** There can be following two situations.

* If the menu has is using the <***select***> tag then you can call the <***selectByValue()>*** or <**selectByIndex()>** or <**selectByVisibleText()>** methods of the Select class.
* **If the menu doesn’t use the <*select*> tag then simply find the XPath of that element and perform the click action for its selection**.

### 2.6.9 Q: How would you select any particular text using the Selenium Webdriver?

**Ans.** It seems an easy one at first but you need to do a little more to achieve this.

|  |
| --- |
| driver.get("/");  WebElement item = driver.findElement(By.xpath("//p[contains(text(),'Selenium webdriver quesions')]"));  Actions dummy = new Actions(driver);  dummy.doubleClick(item).build().perform(); |

### 2.6.10 How to explain Test Automation Framework to the interviewer.

**Must Read:** [Test Automation Framework Interview Questions](https://www.softwaretestingmaterial.com/test-automation-framework-interview-questions/)

“How to explain test automation framework to the interviewer”.

We need to specify in and out of our Test Automation Framework such as:

1. Programming **language** used,
2. **Type of framework** used,
3. Test Base Class (Initializing WebDriver, Implicit Waits)(CommonAPI)
4. How we separate Element locators and tests (Page Objects, Page Factory)
5. Utility functions file
6. Property files
7. TestNG annotations
8. How we parameterize tests using Excel files
9. How we capture error screenshots
10. Generating reports(Extent Reports)
11. Emailing reports
12. Version Control System used
13. Continuous Integration Tool

**Language:** In our Selenium Project we are using Java language. Even though Selenium supports multiple languages, we are using Java language is just because most of the automation developers have knowledge on Selenium with Java.

**Type of Framework:** In our project, we are using [Data-driven Framework](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/) by using [Page Object Model design pattern](https://www.softwaretestingmaterial.com/page-object-model/) with Page Factory.

**Must Read:** [Types of Test Automation Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/)

**POM:** As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. We used Pagefactory in every page Objects, locators of each pages maintained in each web page.

**Packages:** We have separate packages for *Pages* and *Tests*. All the *web page* related classes come under **Pages** package and all the *tests* related classes come under **Tests** package.

For example, *Home Page* and *Login Page* have a separate classes to store element locators. For the *login test* there would be a separate class which calls the methods from the *Home Page* class and *Login Page* class.

**Generic:** This module containsnumber of packages:

1. Base

**CommonAPI.java class**

· Extent Report

· Setup, Initializing the WebDriver, Implicit Waits, Extent Reports Cloud connectivity (eg. Browserstack, Sauce Lab)

**Browserstack Connection:** go to browserstack>resources>capabilities>select as required(eg. Os, browser, version)get capabilities and set it in testNG xml runner file in the framework.

* DesiredCapabilities caps = new DesiredCapabilities();
* caps.setCapability("os", "Windows");
* caps.setCapability("os\_version", "10");
* caps.setCapability("browser", "Chrome");
* caps.setCapability("browser\_version", "62.0");
* caps.setCapability("browserstack.selenium\_version", "3.5.2");

· Common API’s for all other module in order to increase code reusability.

· Stores and handles the functions (The code which is repetitive in nature such as waits, actions, capturing screenshots, accessing excels, sending email etc.,) which can be commonly used across the entire framework. The reason behind creating utility class is to achieve reusability.

2. Reporting

It contains helper methods for Extent Report.

3. Utility

· Database Connection (mongoDB, MySQL)

· Data reader for Microsoft Excel

· Google API to read data from google sheet

We use google Sheets API in order to get authorized connection between test script and google sheet.

#### 2.6.10.1 How to use google sheet?

**Google Sheet:**

Google Sheets are used to pass multiple sets of data to the application. It lets users create and format spreadsheets and simultaneously work with other team members. We used googleAPI to access the Google Sheet. Using google API, code can access and updateGoogle Sheet just like any other user.

Excel allows to run a test case multiple times with different input and assertion value.

We used **Google Developers Console wizard** to create the json file and then store this json file with all the logging credentials under resources directory. Data is read from the spreadsheet using the methods defined in the googleApi class.

Dependency: google-api-client

google-oauth-client-jetty

google-api-services-sheets

#### 2.6.10.2 How to read data from MS Excel?

Purpose to read test data from excel sheet for testing and assertion.

Excel allows to run a test case multiple times with different input and assertion value.

Dependency used: By using org.apache.poi dependency.

How to use:

By using FileInputStream and HSSFWorkbook we implemented a method in generic module by which we can read data from column. Later store this data in array and use it as test data and assertion data.

**Properties file:** This file (***config.properties***) stores the information that remains static throughout the framework such as browser specific information, application URL, screenshots path etc.

All the details which change as per the environment and authorization such as URL, Login Credentials are kept in the *config.properties* file. Keeping these details in a separate file makes easy to maintain.

//In our hybrid framework we maintain it by runner file and locators are being maintained by PageFactory in each page which is our object repository. Instead of maintaining different properties file for locators we use PageFactory @FindBy to maintain locators.

**Screenshots:**  Screenshots will be captured and stored in a separate folder and also the screenshots of a failed test cases will be added in the extent reports.

[How To Capture Screenshots of Failed Test Cases](https://www.softwaretestingmaterial.com/capture-screenshot-of-failed-test-cases-using-selenium-webdriver/)

**Test Data:** All the historical test data will be kept in excel sheet (*controller.xlsx*). By using *‘controller.xlsx’*, we pass test data and handle data driven testing. We use [Apache POI](https://www.softwaretestingmaterial.com/handling-excel-files-using-apache-poi/) to handle excel sheets.

**TestNG:** Using TestNG for Assertions, Grouping and Parallel execution.

Here you could find [TestNG Complete Tutorial](https://www.softwaretestingmaterial.com/testng-tutorial/) and also you could find [TestNG Interview Questions](https://www.softwaretestingmaterial.com/testng-interview-questions/)

**Maven:** Using Maven for build, execution and dependency purpose. Integrating the TestNG dependency in POM.xml file and running this POM.xml file using Jenkins.

[How To Create Maven Project](https://www.softwaretestingmaterial.com/create-selenium-maven-project/)

**Version Control Tool:** We use Git as a repository to store our test scripts.

**Jenkins:** By using Jenkins CI (Continuous Integration) Tool, we execute test cases on daily basis and also for nightly execution based on the schedule. Test Result will be sent to the peers using Jenkins.

[Maven Project with Jenkins](https://www.softwaretestingmaterial.com/execute-maven-project-using-jenkins/)

**Extent Reports:** For the reporting purpose, we are using Extent Reports. It generates beautiful HTML reports. We use the extent reports for maintaining logs and also to include the screenshots of failed test cases in the Extent Report.

Here you could find [How To Generate Extent Reports](https://www.softwaretestingmaterial.com/generate-extent-reports/) and also find [How To Add Screenshots In Extent Reports](https://www.softwaretestingmaterial.com/screenshots-extent-reports/).

You have to explain all these when you are asked to explain test automation framework in the interview. If you have any other thoughts on how to explain test automation framework contact with me I will try to clarify.

**2.7 What is Maven and why it is used?**

When building software projects, you need to perform the following tasks irrespective of the development methodology used.

1. Gather requirements
2. Analyse requirements
3. Design solution
4. Develop (code) solution
5. Test solution
6. Deploy solution
7. Maintain solution

The above tasks are usually executed in the order listed above. For example, you cannot develop a solution before you analyse the requirements and design it. Some may argue that with the Test Driven Development ([Book](http://www.amazon.com/Test-Driven-Development-Kent-Beck/dp/0321146530)), the test cases are created before the solution is developed. While that is true, the tests are still executed after the development is done irrespective of when the tests were developed.

While all the thinking must be done by the team, the repetitive steps can be carried out by an assistant, if we may. The assistant is not intelligent but knows the software development flow and how to do basic things. For example, the assistant can create the projects based on templates, download required third party libraries (also known as JARs) and their dependencies too , build all components and create a deployable JAR, WAR or EAR file. Furthermore, the assistant can be empowered with new skills as the journey proceeds. This assistant is called Maven. This distinguishes Maven from other building tools such as Ant ([Homepage](http://ant.apache.org/)) or other build scripts.

Here I have hand-picked few posts which will help you to learn more interview related stuff:

·  [Selenium Interview Questions](https://www.softwaretestingmaterial.com/selenium-interview-questions/)

·  [Test Automation Framework Interview Questions](https://www.softwaretestingmaterial.com/test-automation-framework-interview-questions/)

·  [TestNG Interview Questions](https://www.softwaretestingmaterial.com/testng-interview-questions/)

·  [SQL Interview Questions](https://www.softwaretestingmaterial.com/sql-interview-questions/)

·  [Manual Testing Interview Questions](https://www.softwaretestingmaterial.com/100-software-testing-interview-questions/)

·  [Agile Interview Questions](https://www.softwaretestingmaterial.com/agile-testing-interview-questions/)

·  [Why You Choose Software Testing As A Career](https://www.softwaretestingmaterial.com/choose-software-testing-as-a-career/)

·  [General Interview Questions](https://www.softwaretestingmaterial.com/6-important-interview-questions/)

## 2.8 Cross Browser Testing Checklist

Here is a [Cross Browser Testing](https://www.softwaretestingmaterial.com/cross-browser-testing) Checklist you could follow while testing your site on different Operating systems and Browsers.

### 2.8.1 Pre-Conditions:

1. List out browsers and operating systems your target audience use.

2. Setup all required browsers for testing.

3. Keep all mobile devices handy or setup emulators or simulators.

Below mentioned items need to test on different Operating Systems like Windows, Linux, and Mac and different browsers (with different versions) like Internet Explorer, Firefox, Google Chrome, Safari, and Opera.

1. Verify the HTML validation – w3.org validator

2. Verify the CSS validation – w3.org js validator

3. Verify the page validations with or without JavaScript enabled

4. Verify the Ajax and JQeury functionality

5. Verify the SSL work correctly in all browsers

6. Verify the alignment of the elements in a web page

7. Verify the spacing between the elements in a web page

8. Verify the font color, font size of the elements in a web page

9. Verify the page layout in different resolutions

10. Verify the spacing between the elements in a web page

11. Verify the styles/effects given are reflecting properly

12. Verify file uploading and file downloading functionality

13. Verify the images are loading after uploading

14. Verify the image alignment and alt tags

15. Verify the text alignment

16. Verify the links in a web page are navigating properly or not

17. Verify the buttons in a web page

18. Verify the popups

19. Verify the videos are working

20. Verify whether the mobile devices are able to scroll up and down the pages properly

21. Verify whether the audio, video and image files are able to upload from devices or desktop

22. Verify the zoom in and out the web pages

23. Verify the Date formats

24. Verify the Tool Tips and mouse hover

25. Verify the Drop-down fields

26. Verify the appearance of horizontal and vertical scrollbar

27. Verify the consistency of animation and flash work

28. Verify the alignment of checkboxes, radio buttons, and forms

29. Verify the Grids/Tables in the page

30. Verify the sessions and cookies

Keep this Cross Browser Testing Checklist handy while doing Cross Browser Testing. If you have any other thoughts, please comment in the comments session below.

## 2.9 Performance Testing

### 2.9.1 JMeter

Before going into the details of JMeter, let us first understand a few jargons associated with the testing of any application.

#### 2.9.1.1 Performance Test −

* This test sets the best possible performance expectation under a given configuration of infrastructure. It also highlights early in the testing process if any changes need to be made before the application goes into production.

#### 2.9.1.2 Load Test −

* This test is basically used for testing the system under the top load it was designed to operate under.

#### 2.9.1.3 Stress Test −

* This test is an attempt to break the system by overwhelming its resources.

#### 2.9.1.4 What is JMeter?

JMeter is a software that can perform load test, performance-oriented business (functional) test, regression test, etc., on different protocols or technologies.

JMeter can conduct load and performance test for many different server types −

Web - HTTP, HTTPS, SOAP,

Database via JDBC, LDAP, JMS,

Mail - POP3, etc.

JMeter store its test plans in XML format. This means you can generate a test plan using a text editor.

It is a platform-independent tool

#### 2.9.1.5 JMeter functioning-

In JMeter, we create the request to the server and simulates number of users and then send the request to server and server responds to it. JMeter saves all the responses and then gather data to calculate statistical info to generate report.

#### 2.9.1.6 What is a Test Plan?

A Test Plan can be viewed as a container for running tests. It defines what to test and how to go about it. A complete test plan consists of one or more elements such as **thread groups, logic controllers, sample-generating controllers, listeners, timers, assertions, and configuration elements**.

**A test plan must have at least one thread group.**

**A plain and blank JMeter window without any additional elements added to it. It contains two nodes −**

* **Test Plan node − is where the real test plan is kept.**
* **Workbench node − It simply provides a place to temporarily store test elements while not in use, for copy/paste purposes. When you save your test plan, Workbench items are not saved with it.**

#### 2.9.1.7 Thread Group

Thread Group elements are the beginning points of your test plan. As the name suggests, the thread group elements control the number of threads JMeter will use during the test. We can also control the following via the Thread Group −

* Setting the number of threads
* Setting the ramp-up time
* Setting the number of test iterations

**The Thread Group Panel holds the following components −**

* Action to be taken after a Sampler error − In case any error occurs during test execution, you may let the test either −
  + Continue to the next element in the test
  + Stop Thread to stop the current Thread.
  + Stop Test completely, in case you want to inspect the error before it continues running.
* Number of Threads − Simulates the number of users or connections to your server application.
* Ramp-Up Period Defines how long it will take JMeter to get all threads running.
* Loop Count − Defines the number of times to execute the test.
* Scheduler checkbox − Once selected, the Scheduler Configuration section appears at the bottom of the control panel.
* Scheduler Configuration − You can configure the start and end time of running the test.

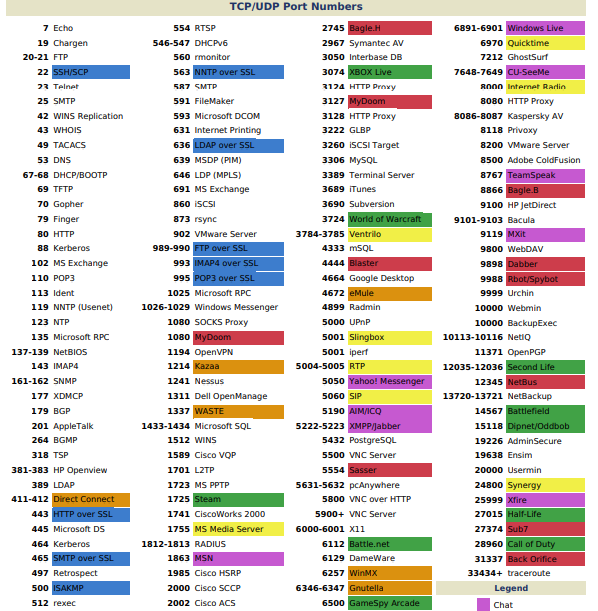
#### 2.9.1.8 Listeners

Listeners let you view the results of Samplers in the form of tables, graphs, trees, or simple text in some log files. They provide visual access to the data gathered by JMeter about the test cases as a Sampler component of JMeter is executed.

Listeners can be added anywhere in the test, including directly under the test plan. They will collect data only from elements at or below their level.

#### 2.9.1.9 Assertions

Assertions allow you to include some validation test on the response of your request made using a Sampler. Using assertions you can prove that your application is returning the correct data. JMeter highlights when an assertion fails.



<http://packetlife.net/media/library/23/common-ports.pdf>

25 SMTP

53 DNS

110 POP3

80 HTTP

443 HTTP over SSL

989-990 FTP over SSL

995 POP3 over SSL

2483-2484 Oracle DB

3306 MySQL

20-21 FTP

[**JMeter**](http://jmeterinterviewquestionsandanswers.blogspot.com/2017/01/35-top-jmeter-interview-questions-with.html) is a JAVA tool, which is used for performance load testing. **JMeter** acts like a group of users sending requests to a target server.

# 3 JMeter Interview Questions and Answers List

### 3.0.1 What is the use of Jmeter?

Jmeter is an open source desktop application software, 100% pure Java application, which is designed to perform load testing of web based application.

Jmeter is used for different types of testing – Performance Testing, Functional Testing, Regression Testing, Unit Testing.

### 3.0.2 For what purpose Jmeter used?

Jmeter is a Performance Tool. It is used to test the performance of the application. Apache JMeter is a 100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

#### 3.0.2.1 What do u sees when u opens a JMeter?

By default,

Test Plan

Workbench

Can be seen

#### 3.0.2.2 Is it mandatory to install java first, to run jmeter?

Yes, Jmeter is an java based application. So you should install java 1.5 or later version in your system.

#### 3.0.2.3 Can we do regression testing with Jmeter?

Yes we can, after-all it is all about java-script.

#### 3.0.2.4 How we can run jmeter?

By simply clicking on Jmeter.bat, or by using commnad: C:\>Jmeter

#### 3.0.2.5 What are the options, do you see when jmtere window opens?

1. Test Plan

2. Workbench

#### 3.0.2.6 What is Test Plan in Jmeter?

Test plan describes a series of steps JMeter will execute when run.

A complete test plan will consist of one or more Thread Groups, logic conrollers, sample generating controllers, listeners, timers, assertions, and configuration elements.

#### 3.0.2.7 What is Work bench?

The Workbench is simply an area to store test elements while you are in the process of constructing a test.

The Workbench is a sandbox for any test or portion of a test that you are working on.

When you are ready to test what you have designed in the Workbench, you can copy or move the elements into the Test Plan.

It also contains Non- Test Elements

Http mirror sever

Http Proxy server

which is not available in the thread group & Test plan

#### 3.0.2.8 Which component, do we need to add, to record script?

We need to add Http proxy server, by simply right clicking on Workbench > Non-Test elements > Http proxy server.Also need to enable proxy, with port 8080, by going to internet settings> Connections > Lan. Name will be "localhost".Then click on start button on jmeter http proxy server window.

#### 3.0.2.9 Give a brief history about Jmeter?

A Brief History of Jmeter:

Stefano Mazzocchi is credited for being the developer of this unique tool- Apache JMeter, which is claimed to have been developed to test the strength and performance of the Apache JServ. However; it was later redesigned to make it capable of doing, performance testing, functional testing, regression testing, and unit testing as well.

Since November 2011, the Apache JMeter project has been included in the group of the company’s Top Level projects and today it consists of a separate website and a dedicated Project Management Committee to handle its operations and to provide assistance.

#### 3.0.2.10 Can we perform load testing of database with jmeter?

Yes we can. To see the process follow: Database load testing with jmeter link. You should be aware of sql queries.

#### 3.0.2.11 What are listeners in Jmeter?

Listeners in jmeter are used to capture the response data of load test. We can have response data (result) in , cvs, image (graph) or in html format.

#### 3.0.2.12 What are the benefits that JMeter offers for performance testing?

JMeter offers benefits on performance testing like,

• It can be used to test performance for both, static resources as well as dynamic resources

• It can handle a maximum number of concurrent users then your website can handle

• It provides the graphical analyses of performance reports

#### 3.0.2.13 Name 4-5 listeners in jmeter, which are commonly used.

1. View result tree

2. Summary report

3. Aggregate Graph

4. Response time vs thread

5. Composite graph

#### 3.0.2.14 What is Response time in Jmeter report?

Response time in Jmeter report, is the time taken by web server, to answer the requested query. In simple words, it is the time taken by a web server to respond the http request.

#### 3.0.2.15 What is Spike testing and how can we perform it in JMeter?

Suddenly spiking or increasing the number of users at certain point of application and then monitoring the behavior at that intervals is spike testing.

In JMeter spike testing can **be achieved using Synchronizing Timer**. Synchronizing timer blocks the threads until a particular number of threads have been blocked, and then release them at once thus creating large instantaneous load.

#### 3.0.2.16 What is median in jmeter listener?

It is a number which divides the samples into two equal halves. Half of the samples are smaller than the median, and half are larger.

#### 3.0.2.17 Explain how JMeter works?

JMeter acts like a group of users sending requests to a target server. It collects response from target server and other statistics which show the performance of the application or server via graphs or tables.

#### 3.0.2.18 What is throughput in jmeter listener?

Throughput is relation between time and data.It shows the relation between http request and unit time. Means how many requests are processed in how much time.

**Throughput = Number of requests / Time(second or millisecond)**

#### 3.0.2.19 What are the other applications tested by Jmeter?

JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more)

#### 3.0.2.20 Which Jmeter elements are used for Parameterization?

1. User parameter

2. CSV Data Set config

#### 3.0.2.21 Which controllers are used as an conditional controllers in jmeter?

If controller and While controller

#### 3.0.2.22 What are listeners in Jmeter?

Listeners in Jmeter are used to capture the response data of load test. We can have response data (result) in cvs, image (graph) or in html format.

#### 3.0.2.23 What is heap size in jmeter?

Jmeter is work in java, and heap size is related to java. Java heap is the heap size allocated to JVM applications which takes care of the new objects being created. If the objects being created exceed the heap size, it will throw an **error of java.lang.out.of.memory.**

Java's default heap size limit in Jmeter is 256 MB.

27. What is median in jmeter listener?

It is a number which divides the samples into two equal halves. Half of the samples are smaller than the median, and half are larger.

28. Is it possible that, if we increase the heap size in Jmeter, then application(jmeter) might not work?

Yes, it is possible. Because heap size depends upon the empty space of your machine in c drive as well as your RAM. If the RAM is 1 Gb and you increased the heap 2048, it may not work.

29. Explain parameterization in JMeter?

Parameterization is process of generalizing some user input, so as to use it for multiple users or executions.

30. Which Jmeter elements are used for Parameterization?

• User parameter

• CSV Data Set config

31. Explain what a timer in JMeter is and what are the types of it?

A JMeter thread by default will send requests continuously without any pause. To get a pause between the request, Timers are used. Some of the Timers used are Constant Timer, Gaussian Random Timer, Synchronizing Timer, and Uniform Random Timer and so on.

#### 3.0.2.24 Explain what is the role of Timer in JMeter?

With the help of timer, JMeter can delay the time between each request, which a thread makes. It can solve the overload problem of the server.

#### 3.0.2.25 Explain what is Assertion in JMeter? What are the types of assertion?

Assertion helps to verify that your server under test returns the expected results.

Some commonly used Assertion in JMeter are:

• Response Assertion

• Duration Assertion

• Size Assertion

• XML Assertion

• HTML Assertion

#### 3.0.2.26 What is Blaze Meter?

Blaze Meter is a cloud based service compatible with Apache JMeter. It generates large amount of instant load and provide very comprehensive reporting and analysis features. In Blaze meter we can just upload the JMeter script and run the load test on cloud with predefined number of users.

#### 3.0.2.27 How do you ensure re-usability in your JMeter scripts?

· Using config elements like "CSV Data Set Config", "User Defined Variables", etc for greater data reuse

· Modularizing shared tasks and invoking them via a "Module Controller".

· Writing your own BeanShell functions, and reusing them.

# 4 Appium

### **4.0.1 JSON wire protocol (JSONWP)**

The **JSON wire protocol** (**JSONWP**) is a transport mechanism created by WebDriver developers. This wire protocol is a specific set of predefined, standardized endpoints exposed via a RESTful API. The purpose of WebDriver and JSONWP is the automated testing of websites via a browser such as Firefox driver, IE driver, and Chrome driver.

Appium implements the Mobile JSONWP, the extension to the Selenium JSONWP, and it controls the different mobile device behaviors, such as installing/uninstalling apps over the session.

### 4.0.2 What is Appium

**Appium** is an *open source* test automation tool for mobile applications. It allows you to test all the three types of mobile applications: *native*, *hybrid* and *mobile web*.

It also allows you to run the automated tests on **actual devices, emulators and simulators.**

Today when every mobile app is made in at least two platform iOS and Android, you for sure need a tool, which allows testing cross platform. Having two different frameworks for the same app increases the cost of the product and time to maintain it as well.

**The basic philosophy of Appium is that you should be able to reuse code between iOS and Android,** and that’s why when you see the API they are same across iOS and android. Another important thing to highlight here is that unlike Calabash, Appium doesn’t modify your app or need you to even recompile the app.

Appium let’s you choose the language you want to write your test in. It doesn’t dictate the language or framework to be used.

### 4.0.3 Architecture

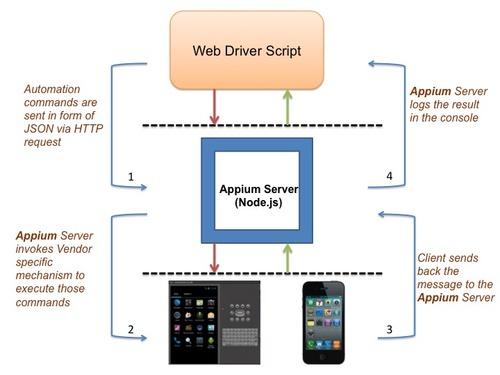
So how does Appium do all this? Let’s try to understand what happens behind the scenes.

When you download the Appium app you are basically downloading the server. The server is written in Node.js and implements the Selenium WebDriver. It allows one to use available WebDriver client to fire your tests. And then your mobile app acts precisely like a web app where the DOM is represented by View hierarchy.

So this server basically exposes REST api which performs the following actions:

1) Receives connection from client  
2) Listen command  
3) Execute command  
4) Respond back the command execution status

In terms of architecture diagram, below is how Appium can be explained.



**Appium Interview Questions**

1) What is Appium?

• Appium is an open-source tool for automating Native, Mobile Web, and Hybrid applications on iOS and Android platforms.

• Appium is "cross-platform": it allows us to write tests against multiple platforms (iOS, Android), using the same API. This enables code reuse between iOS and Android testsuites.

• Appium was launched in 2012.

2) What are Native Apps?

Native apps are those written using the iOS or Android SDKs.

3) What are Mobile Web Apps?

Mobile web apps are web apps accessed using a mobile browser.

Note:

4) What are Hybrid Apps?

Hybrid apps have a wrapper around a "webview" -- a native control that enables interaction with web content.

5) Which language is used to write tests in Appium?

• We can write our tests in any language, Since Appium is nothing more than an HTTP server, a test which needs to be interfaced with Appium can simply use HTTP libraries to create HTTP sessions.

• You just need to know the Selenium protocol in order to compose the right commands and that's it!

• Most common languages and development frameworks are: C#.NET, Java, Ruby, Python and Javascript.

6) What are the pre-requisite to use APPIUM?

Pre-requisite to use APPIUM is

• ANDROID SDK

• Eclipse IDE

• JDK

• Selenium Server JAR

• Webdriver Language Binding Library

• TestNG

• APPIUM for Windows

• APK App Info On Google Play

• js

7) What are the Advantages of Appium?

• Appium allows us to write tests against multiple mobile platforms using the same API.

• We can write and run our tests using any language or test framework.

• It is an open-source tool that we can easily contribute to.

• Provides cross-platform for Native and Hybrid mobile automation.

• Support JSON wire protocol.

• It does not require recompilation of App.

• Support automation test on physical device as well as similar or emulator both.

• It has no dependency on mobile device.

8) What are the limitations of Appium?

• Appium does not support testing of Android Version lower than 4.2

• Limited support for hybrid app testing. E.g., not possible to test the switching action of application from the web app to native and vice-versa.

• No support to run Appium Inspector on Microsoft Windows.

9) What test frameworks are supported by Appium?

• Appium does not support test frameworks because there is no need to support them.

• We can use Appium with all test frameworks we want.

• NUnit and .NET Unit Test Framework are just a few examples.

• We will write our tests using one of the drivers for Appium; thus our tests will interface with Appium just in terms of an external dependency.

10) What are the basic requirements to write Appium tests?

To write Appium tests we require:

• Driver Client: Appium drives mobile applications as though it were a user. Using a client library you write your Appium tests which wrap your test steps and sends to the Appium server over HTTP.

• Appium Session: You have to first initialize a session, as such Appium test takes place in the session. Once the Automation is done for one session, it can be ended and wait for another session

• Desired Capabilities: To initialize an Appium session you need to define certain parameters known as “desired capabilities” like PlatformName, PlatformVersion, Device Name and so on. It specifies the kind of automation one requires from the Appium server.

• Driver Commands: You can write your test steps using a large and expressive vocabulary of commands.

11) What is Appium Inspector?

• Similar to Selenium IDE record and Playback tool, Appium has an “Inspector” to record and playback.

• It records and plays native application behavior by inspecting DOM and generates the test scripts in any desired language.

• However, Appium Inspector does not support Windows and use UIAutomator viewer in its option.

12) How can we exchange data between our test and the App we are testing?

• Appium, actually the WebDriver specification, is not made for exchanging data with our app.

• Actually it is not impossible to exchange data with our app , however it will require us to build more layers of testability.

13) Explain the design of Appium?

• Appium is an “HTTP Server” written using Node.js platform and drives iOS and Android session using Webdriver JSON wire protocol.

• When Appium is downloaded and installed, then a server is setup on our machine that exposes a REST API.

• Appium receives connection and command request from the client and execute that command on mobile devices (Android / iOS).

• Appium responds back with HTTP responses. Again, to execute this request, it uses the mobile test automation frameworks to drive the user interface of the apps.

Mobile Test Automation Frameworks are:

Apple Instruments for iOS.

Google UIAutomator for Android API level 16 or higher.

Selendroid for Android API level 15 or less.

# 5 Programming Language

## 5.1 Core Java

### 5.1.1 What jdk contains or composed of and define each components?

Answer: Components of JDK:

1. **The Runtime Interpreter:** The Java runtime interpreter (java) is a stand-alone version of the Java interpreter built into the HotJava browser. The runtime interpreter provides the support to run Java executable programs in compiled, bytecode format.

2. **The Compiler**: The Java compiler (javac) is used to compile Java source code files into executable Java bytecode classes. In Java, source code files have the extension .java.

3. **The Applet Viewer**: The applet viewer is a tool that serves as a minimal test bed for final release Java applets. You can use the applet viewer to test your programs instead of having to wait for HotJava to support the final release of Java.

Explanation: A **Java applet** is a [small application](https://en.wikipedia.org/wiki/Applet) that is written in the [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) programming language, or another [programming language](https://en.wikipedia.org/wiki/Programming_language) that compiles to [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode), and delivered to users in the form of Java [bytecode](https://en.wikipedia.org/wiki/Bytecode). The user launches the Java applet from a [web page](https://en.wikipedia.org/wiki/Web_page), and the applet is then executed within a [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) in a [process](https://en.wikipedia.org/wiki/Process_(computing)) separate from the [web browser](https://en.wikipedia.org/wiki/Web_browser) itself. A Java applet can appear in a frame of the web page, a new application window, [Sun](https://en.wikipedia.org/wiki/Sun_Microsystems)'s [AppletViewer](https://en.wikipedia.org/wiki/AppletViewer), or a stand-alone tool for testing applets.

4. **The Debugger:** Debugger does the job of debugging the a piece of code.

5. **The Class File Disassembler:** Disassembles class files.

6. **The Header and Stub File Generator:** Generates header and source files that are needed to implement native methods.

7. **The Documentation Generator:** Generates HTML pages of API documentation from Java source files.

The Java documentation generator (javadoc) is a useful tool for generating API documentation directly from Java source code. The documentation generator parses through Java source files and generates HTML pages based on the declarations and comments.

### 5.1.2 What is IDE?

Answer: An IDE stands Integrated Development Environment is a software suite that consolidates the

basic tools developers need to write and test software. Typically, an IDE contains a code editor, a

compiler or interpreter and a debugger that the developer accesses through a single graphical user

interface (GUI).

### 5.1.3 What are the IDEs available in the market to support java?

Answer: Eclipse, Intellij Idea, NetBeans, AndroidStudio, Enide Studio 2014, BlueJ, jEdit,

jGRASP, JSource, JDeveloper, DrJava etc.

### 5.1.4 Explain the entire java life cycle.

Answer: Programmer use any text editor to create high level Java text file.This file is saved as a .java file on the disk.Then compile this text file using the Java compiler, which result

in a .classfile being created on the disk. The .class file contains the bytecodes.

The file is then loaded into memory by the class loader. The bytecode verifier confirms that the bytecodes are valid and not hostile. Finally, the JVM reads the bytecodes in memory and translates them into machine code

### 5.1.5 What is class?

Answer: Class is a blueprint from which individual objects are created.

Classes have data and methods, data represents state and methods behavior. Class is just a logical framework.

So, a class is a template for an object and object is an instance of a class

### 5.1.6 What is object?

Answer: Object is an instance of the class and Object is the physical reality.

### 5.1.7 What is the entry point of an application?

Answer: Main method is the entry point of an application.

### 5.1.8 Why main is static?

In order to call a method need to have a object of that method. As main method

is gateway to the java application JVM need to enter into main method in order

to execute. Static keyword let JVM to call main method without object.

### 5.1.9 What is Abstract Method?

An **abstract method** is a **method** that is declared, but contains no implementation.

Interfaces describes specification (or behavior) of a component.

Example: public abstract void makeNoise();

### 5.1.10 What is the difference between an Interface and Abstract class?

Answer: A class that contains **one or more abstract methods** , and therefore can never

be instantiated. **Abstract classes** are defined so that other classes can

extend them and make them concrete by implementing the abstract methods.

**An interface** in the Java programming language is an abstract type

that is used to **specify a behaviour that classes must implement.**

**An interface is a blueprint of a class.**

Interfaces are declared using the Interface keyword, and may only

contain method signature and constant declarations(variable declarations that

are declared to be both static and final).

### 5.1.11 Can you make an Object from an Interface and Abstract class ? if not how do you use it ?

No we can not create object in Interface and Abstract class.

Interface contains methods to be implemented, in abstract class we can implement some of them and can create more new methods as required keeping at least one abstract method. later abstract class can be extended and create concrete class where there will be no abstract methods.

### 5.1.12 What is Access Specifier?

Access Specifier is to determine the visibility of methods and variables.Access Specifier set the accessibility of classes, methods, and other members. Access modifiers are a specific part of programming language syntax used to facilitate the [encapsulation](https://en.wikipedia.org/wiki/Encapsulation_(computer_programming)) of components.

Here is a summary of the four access modifiers in Java that control visibility:

1. Private is Visible to the class only (private).

2. Public is Visible to the world (public).

3. Protected is Visible to the package and all subclasses (Protected).

4. Visible to the package—the (unfortunate) default. No modifiers are needed.

A subclass cannot access the private fields of its superclass.

In practice, use protected fields with caution.

### 5.1.13 What is OOP ? Define each feature of java OOP.

OOP. Stands for "Object-Oriented Programming." Refers to a programming methodology based on objects, instead of just functions and procedures. These objects are organized into classes, which allow individual objects to be group together.

Features of java OOP:

1. Encapsulation

2. Inheritance

3. Polymorphism

4. Abstraction

#### 

#### 5.1.13.1 Encapsulation:

Encapsulation **means data hiding**. By doing encapsulation data can only be be accessed by public methods. There are various way to hide the data. By making a method, variable

private access limits to the class user. **By using getter and setter method private data can be accessed, hiding implementation of methods and data from user is called encapsulation.** It improves maintainability and flexibility and reusability. The fields can be made read-only. User would not be knowing what is going on behind the scene. They would only be knowing that to update a field call set method and to read a field call get method but what these set and get methods are doing is purely hidden from them.

#### 5.1.13.2 Inheritance:

Inheritance refers to a feature of Java programming that lets you create

classes that are derived from other classes. A class that's based on another class inherits

the other class. The class that is inherited is the parent class, the base class, or the superclass.

Java does not support multiple inheritance.

#### 5.1.13.3 Polymorphism:

Polymorphism is the ability of an object to take on many forms. The most common

use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object. Any Java object that can pass more than one IS-A test is considered to be polymorphic.

Polymorphism in Java has two types:

a. Compile time polymorphism (static binding):It happens during compile time.

Example: Method overloading is an example of static polymorphism,

b. Runtime polymorphism (dynamic binding): It happens during run time.

Example: Method overriding is an example of dynamic polymorphism.

#### 5.1.13.4 Abstraction:

Abstraction in Java or Object oriented programming is a way to segregate

implementation from an interface and one of the five fundamentals along with Encapsulation, Inheritance, Polymorphism, Class, and Object. **Abstraction in Java is achieved by using interface and abstract class in Java.** An interface or abstract class is something which is not concrete , something which is incomplete. In order to use interface or abstract class, we need to extend and implement an abstract method with concrete behavior.

### 5.1.14 What is API? Name list of API that you have used so far.

An application program interface (API) is a set of routines, protocols, and tools for building

software applications. Basically, an API specifies how software components should interact.

Additionally, APIs are used when programming graphical user interface (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.

### 5.1.15 Does java support multiple inheritance?

Answer: No, but Java supports **multilevel** inheritance and **multiple Interfaces**.

### 5.1.16 What is method overloading and when it happens?

Answer: When same method takes different signature it is called method overloading

it happen **within class**. It is static polymorphism and happens during compile time.

|  |
| --- |
| *//A class for adding upto 5 numbers* class Sum {  int add(int n1, int n2)   {  return n1+n2;  }  int add(int n1, int n2, int n3)   {  return n1+n2+n3;  }  int add(int n1, int n2, int n3, int n4)   {  return n1+n2+n3+n4;  }  int add(int n1, int n2, int n3, int n4, int n5)   {  return n1+n2+n3+n4+n5;  }  public static void main(String args[])  {  Sum obj = new Sum();  System.out.println("Sum of two numbers: "+obj.add(20, 21));  System.out.println("Sum of three numbers: "+obj.add(20, 21, 22));  System.out.println("Sum of four numbers: "+obj.add(20, 21, 22, 23));  System.out.println("Sum of five numbers: "+obj.add(20, 21, 22, 23, 24));  } } |

### 5.1.17 Overriding

Ability to define a behavior that's specific to the subclass type, which means a subclass can implement a parent class **method** based on its requirement.

In object-oriented terms, **overriding** means to **override** the functionality of an existing **method**. It is dynamic polymorphism and happens during runtime.

|  |
| --- |
| package beginnersbook.com; class CarClass {  public int speedLimit()   {  return 100;  } } class Ford extends CarClass {  public int speedLimit()  {  return 150;  }  public static void main(String args[])  {  CarClass obj = new Ford();  int num= obj.speedLimit();  System.out.println("Speed Limit is: "+num);  } } |

### 5.1.18 Explain exceptions in java and how to handle it.

Answer: The term exception is shorthand for the phrase "exceptional event."An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

#### 5.1.18.1 Errors

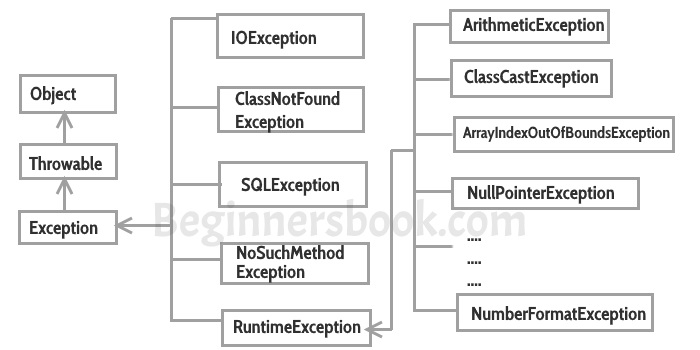
indicate that something severe enough has gone wrong, the application should crash rather than try to handle the error.

#### 5.1.18.2 Exceptions

are events that occurs in the code. A programmer can handle such conditions and take necessary corrective actions.

#### 5.1.18.3 Types of exceptions

There are two types of exceptions in Java:

1)Checked exceptions/ Detected at compile time 

2)Unchecked exceptions/ Detected at runtime

### 5.1.19 How to use Try Catch Block?

The try block contains set of statements where an exception can occur. A try block is always followed by a catch block, which handles the exception that occurs in associated try block. A try block must be followed by catch blocks or finally block or both.

|  |
| --- |
| try {  *//statements that may cause an exception* } catch (exception(type) e(object))‏ {  *//error handling code* } |

|  |
| --- |
| **catch**(Exception e){  //This catch block catches all the exceptions } |

A catch block is where you handle the exceptions, this block must follow the try block. A single try block can have several catch blocks associated with it. You can catch different exceptions in different catch blocks. When an exception occurs in try block, the corresponding catch block that handles that particular exception executes. For example if an arithmetic exception occurs in try block then the statements enclosed in catch block for arithmetic exception executes.

|  |
| --- |
| class Example1 {  public static void main(String args[]) {  int num1, num2;  try {  */\* We suspect that this block of statement can throw   \* exception so we handled it by placing these statements  \* inside try and handled the exception in catch block  \*/*  num1 = 0;  num2 = 62 / num1;  System.out.println(num2);  System.out.println("Hey I'm at the end of try block");  }  catch (ArithmeticException e) {   */\* This block will only execute if any Arithmetic exception   \* occurs in try block  \*/*  System.out.println("You should not divide a number by zero");  }  catch (Exception e) {  */\* This is a generic Exception handler which means it can handle  \* all the exceptions. This will execute if the exception is not  \* handled by previous catch blocks.  \*/*  System.out.println("Exception occurred");  }  System.out.println("I'm out of try-catch block in Java.");  } } |

The example we seen above is having multiple catch blocks, lets see few rules about multiple catch blocks with the help of examples. To read this in detail, see [catching multiple exceptions in java](https://beginnersbook.com/2013/05/catch-multiple-exceptions/).

1. As I mentioned above, a single try block can have any number of catch blocks.

2. A generic catch block can handle all the exceptions. Whether it is ArrayIndexOutOfBoundsException or ArithmeticException or NullPointerException or any other type of exception, this handles all of them. To see the examples of NullPointerException and ArrayIndexOutOfBoundsException, refer this article: [Exception Handling example programs](https://beginnersbook.com/2013/04/exception-handling-examples/).

|  |
| --- |
| catch(Exception e){  *//This catch block catches all the exceptions* } |

If you are wondering why we need other catch handlers when we have a generic that can handle all. This is because in generic exception handler you can display a message but you are not sure for which type of exception it may trigger so it will display the same message for all the exceptions and user may not be able to understand which exception occurred. Thats the reason you should place is at the end of all the specific exception catch blocks

3. If no exception occurs in try block then the catch blocks are completely ignored.

4. Corresponding catch blocks execute for that specific type of exception:

catch(ArithmeticException e) is a catch block that can hanlde ArithmeticException

catch(NullPointerException e) is a catch block that can handle NullPointerException

5. You can also throw exception, which is an advanced topic and I have covered it in separate tutorials: [user defined exception](https://beginnersbook.com/2013/04/user-defined-exception-in-java/), [throws keyword](https://beginnersbook.com/2013/12/throws-keyword-example-in-java/), [throw vs throws](https://beginnersbook.com/2013/04/difference-between-throw-and-throws-in-java/).

### **5.1.20 Example of Multiple catch blocks**

|  |
| --- |
| class Example2{  public static void main(String args[]){  try{  int a[]=new int[7];  a[4]=30/0;  System.out.println("First print statement in try block");  }  catch(ArithmeticException e){  System.out.println("Warning: ArithmeticException");  }  catch(ArrayIndexOutOfBoundsException e){  System.out.println("Warning: ArrayIndexOutOfBoundsException");  }  catch(Exception e){  System.out.println("Warning: Some Other exception");  }  System.out.println("Out of try-catch block...");  } } Output: Warning: ArithmeticException Out of try-catch block... |

### 5.1.21 What is static keyword in java? How it has been used in variables and methods ?

Answer: The he static keyword denotes that a member variable, or method, can be accessed without requiring

an instantiation of the class to which it belongswhen static key word is used with a method that method can

be called without creating a object.

### 5.1.22 What is final and how it has been used variables and methods?

A final class cannot be subclassed. A final variable can only be initialized once.

A final method cannot be overridden by subclasses.

### 5.1.23 What is final, finally and finalize?

Final:

Final variables are nothing but constants. We cannot change the value of a final variable once it is initialized. A final method cannot be overridden. Which means even though a sub class can call the final method of parent class without any issues but it cannot override it.

Finally:

Finally key word used to execute finally block when the try and catch block exits which ensures that the finally block is executed even if an unexpected exception occurs.

Finalize():

Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **final** | **finally** | **finalize** |
| 1) | Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed. | Finally is used to place important code, it will be executed whether exception is handled or not. | Finalize is used to perform clean up processing just before object is garbage collected. |
| 2) | Final is a keyword. | Finally is a block. | Finalize is a method. |

<https://beginnersbook.com/2013/04/java-exception-handling/>

<https://www.geeksforgeeks.org/g-fact-24-finalfinally-and-finalize-in-java/>

#### 5.1.23.1 Finally:

Just as final is a reserved keyword, so in same way **finally is also a reserved keyword in java** i.e, we can’t use it as an identifier. The finally keyword is used in association with a [try/catch block](https://www.geeksforgeeks.org/flow-control-in-try-catch-finally-in-java/) and guarantees that a section of code will be executed, even if an exception is thrown. The finally block will be executed after the try and catch blocks, but before control transfers back to its origin.

#### 5.1.23.2 try-catch-finally block

* Either a try statement should be associated with a catch block or with finally.
* Since catch performs exception handling and finally performs the cleanup, the best approach is to use both of them.

**Example 1:** The following example demonstrate the working of finally block when no exception occurs in try block

|  |
| --- |
| class Example1{  public static void main(String args[]){  try{  System.out.println("First statement of try block");  int num=45/3;  System.out.println(num);  }  catch(ArrayIndexOutOfBoundsException e){  System.out.println("ArrayIndexOutOfBoundsException");  }  finally{  System.out.println("finally block");  }  System.out.println("Out of try-catch-finally block");  } } Output: First statement of try block 15 finally block Out of try-catch-finally block |

**Example 3**: When exception occurs in try block and handled properly in catch block

|  |
| --- |
| class Example3{  public static void main(String args[]){  try{  System.out.println("First statement of try block");  int num=45/0;  System.out.println(num);  }  catch(ArithmeticException e){  System.out.println("ArithmeticException");  }  finally{  System.out.println("finally block");  }  System.out.println("Out of try-catch-finally block");  } } Output: First statement of try block ArithmeticException finally block Out of try-catch-finally block |

#### **5.1.23.3 Application of finally block:**

So basically the use of finally block is **resource deallocation**. Means all the resources such as **Network Connections, Database Connections,** which we opened in try block are needed **to be closed**, so that we won’t lose our resources as opened. So those resources are needed to be closed in finally block.

### 5.1.24 Finalize method

It is a **method** that the [**Garbage Collector**](https://www.geeksforgeeks.org/garbage-collection-java/) always calls just **before** the deletion/destroying the object which is eligible for Garbage Collection, so as to perform **clean-up activity**. Clean-up activity means closing the resources associated with that object like Database Connection, Network Connection or we can say resource de-allocation. Remember it is **not** a reserved keyword.

Once finalize method completes immediately Garbage Collector destroy that object. finalize method is present in Object class and its syntax is:

**protected void finalize throws Throwable{}**

Since Object class contains finalize method hence finalize method is available for every java class since Object is superclass of all java classes. Since it is available for every java class hence Garbage Collector can call finalize method on **any java object**

Now, the finalize method which is present in Object class, has empty implementation, in our class clean-up activities are there, then we have to **override** this method to define our own clean-up activities.

Cases related to finalize method:

1. **Case 1 :** The object which is eligible for Garbage Collection, that object’s corresponding class finalize method is going to be executed

|  |
| --- |
|  |

|  |
| --- |
| class Hello {  public static void main(String[] args)  {  String s = new String("RR");  s = null;    *// Requesting JVM to call Garbage Collector method*  System.gc();  System.out.println("Main Completes");  }    *// Here overriding finalize method*  public void finalize()  {  System.out.println("finalize method overriden");  } } Run on IDE Output: Main Completes |

**Note** : Here above output came only **Main Completes** and **not** “finalize method overriden” because Garbage Collector call finalize method on that class object which is eligible for Garbage collection. Here above we have done->

**s = null** and ‘s’ is the object of String class, so String class finalize method is going to be called and not our class(i.e, Hello class). So we modify our code like->

1. **Hello s = new Hello()**;  
   **s = null**;
2. Now our class i.e, Hello class finalize method is called. **Output**:
3. finalize method overriden  
   Main Completes
4. So basically, Garbage Collector calls finalize method on that class object which is eligible for Garbage collection.So if String object is eligible for Garbage Collection then **String** class finalize method is going to be called and **not the Hello class** finalize method.
5. **Case 2 :** We can call finalize method Explicitly then it will be executed just like normal method call but object won’t be deleted/destroyed

|  |
| --- |
|  |

|  |
| --- |
| class Bye {  public static void main(String[] args)  {  Bye m = new Bye();    *// Calling finalize method Explicitly.*  m.finalize();  m.finalize();  m = null;    *// Requesting JVM to call Garbage Collector method*  System.gc();  System.out.println("Main Completes");  }    *// Here overriding finalize method*  public void finalize()  {  System.out.println("finalize method overriden");  } } Run on IDE Output: finalize method overriden  *//call by programmer but object won't gets destroyed.* finalize method overriden  *//call by programmer but object won't gets destroyed.* Main Completes finalize method overriden  *//call by Garbage Collector just before destroying the object.* |

1. **Note** : As finalize is a method and not a reserved keyword, so we can call finalize method **Explicitly**, then it will be executed just like normal method call but object won’t be deleted/destroyed.
2. **Case 3 :**
   * **Part a)** If programmer calls finalize method, while executing finalize method some unchecked exception rises.

|  |
| --- |
|  |

|  |
| --- |
| class Hi {  public static void main(String[] args)  {  Hi j = new Hi();    *// Calling finalize method Explicitly.*  j.finalize();    j = null;    *// Requesting JVM to call Garbage Collector method*  System.gc();  System.out.println("Main Completes");  }    *// Here overriding finalize method*  public void finalize()  {  System.out.println("finalize method overriden");  System.out.println(10 / 0);  } } Run on IDE Output: exception in thread "main" java.lang.ArithmeticException: / by zero followed by stack trace. |

* + So **key point** is : If programmer calls finalize method, while executing finalize method some unchecked exception rises, then JVM terminates the program abnormally by rising exception. So in this case the program termination is **Abnormal**.
  + **part b)** If garbage Collector calls finalize method, while executing finalize method some unchecked exception rises.

|  |
| --- |
|  |

|  |
| --- |
| class RR {  public static void main(String[] args)  {  RR q = new RR();  q = null;    *// Requesting JVM to call Garbage Collector method*  System.gc();  System.out.println("Main Completes");  }    *// Here overriding finalize method*  public void finalize()  {  System.out.println("finalize method overriden");  System.out.println(10 / 0);  } }  Output: Main Completes finalize method overriden |

So **key point** is : If Garbage Collector calls finalize method, while executing finalize method some unchecked exception rises then JVM **ignores** that exception and rest of program will be continued normally. So in this case the program termination is **Normal**and not abnormal.

**Important points:**

* There is no guarantee about the time when finalize is called. It may be called any time after the object is not being referred anywhere (cab be garbage collected).
* JVM does not ignore all exceptions while executing finalize method, but it ignores **only** [**Unchecked exceptions**](https://www.geeksforgeeks.org/checked-vs-unchecked-exceptions-in-java/). If the corresponding catch block is there then JVM won’t ignore and corresponding catch block will be executed.
* System.gc() is just a request to JVM to execute the Garbage Collector. It’s up-to JVM to call Garbage Collector or not.Usually JVM calls Garbage Collector when there is not enough space available in the Heap area or when the memory is low.

### 5.1.25 What is a constructor ?

Constructor is being used to instantiate a object by using new operator. It is an instance method with

the same name of their class.

### 5.1.26 What is "this" keyword in java ?

this key word represents instance of the class. this can be used to access class methods and variables.

### 5.1.27 What is "super" keyword in java?

Answer: super key word used for accessing the members of the super class by the extended class.

### 5.1.28 What is JVM stand for ?

Answer: Java Virtual Machine

### 5.1.29 What version of java are you using?

Answer: java version "9.0.1"

Java(TM) SE Runtime Environment (build 9.0.1+11)

Java HotSpot(TM) 64-Bit Server VM (build 9.0.1+11, mixed mode)

### 5.1.30 What is JAR stand for ?

Answer: Java Archive is a platform independent file format that holds many files into one.

Multiple applets written in the Java Programming language and their components

like .class files, images, sounds and other resource files can be bundled in a JAR file and

downloadable in a single HTTP transaction. It supports file compression and digital signatures.

### 5.1.31 What is compile time and run time?

Java compile the .java file to .class file, this stage is called compile time.

java loader loads the .class file in JVM and JVM executes the application.

This execution time of a application is called run time.

### 5.1.32 What is heap?

Answer: The heap is the run-time data area from which memory for all class instances and arrays is allocated.

Heap is the place where all objects stays during run time. It is part of RAM.

### 5.1.33 How java manage it's memory?

Answer: All the primitive variables and reference variable stays in stack. Heap contains the objects.

Main methods executes on stack. during the process of execution when a object lost his reference variable

which is being erased from the stack after completion of execution is waits for the Java automatic garbage

collector, and object without no reference variable being removed from heap by garbage collector.

### 5.1.34 What is the difference between String, StringBuffer and StringBuilder?

Answer: String is immutable, if you try to alter their values, another object gets created, whereas StringBuffer and StringBuilder are mutable so they can change their values.StringBuffer methods are synchronized while StringBuilder methods are non-synchronized, it means that for thread-safe operations you must choose StringBuffer class instead of StringBuilder.

The StringBuffer and StringBuilder classes are used when there is a necessity to make a lot of modifications to Strings of characters.

Unlike Strings, objects of type StringBuffer and StringBuilder can be modified over and over again without leaving behind a lot of new unused objects.

### 5.1.35 What is Singleton class?

Answer: The purpose of singleton is to control object creation by **keeping private constructor**. **Singleton pattern helps us to keep only one instance of a class at any time**.We can make constructor as private. So that user can not create an object from outside of the class.

In **Java** the **Singleton pattern** will ensure that there is only one instance of a **class** is created in the **Java** Virtual Machine. It is used to provide global point of access to the object. In terms of practical **use Singleton** patterns are used in logging, caches, thread pools, configuration settings, device driver objects.

### 5.1.36 What is Serialization and Deserialization?

Answer: **Serialization**: It is a process of converting an object into a sequence of bytes which can be persisted to a disk or database or can be sent through streams.

**Deserialization:** The reverse process of creating object from sequence

of bytes is called deserialization.

### 5.1.37 Which one will take more memory, an int or Integer?

Answer: An int shall take 4 bytes and Integer shall take about 16 bytes space in heap.Integer are wrapper class which has more overhead compared to primitive type. The memory space consumption by object depends upon JVM during run time environment. For a object this space varies.

### 5.1.38 Why is String Immutable in Java?

Answer: **Security:** parameters are typically represented as String in network connections, database connection urls, usernames/passwords etc. If it were mutable, these parameters could be easily changed. Synchronization and concurrency: making String immutable automatically makes them thread safe thereby solving the synchronization issues.

**Caching**: when compiler optimizes your String objects, it sees that if two objects have same value (a="test", and b="test") and thus you need only one string object (for both a and b, these two will point to the same object).

**Class loading**: String is used as arguments for class loading. If mutable, it could result in wrong class being loaded (because mutable objects change their state).

### 5.1.39 What is constructor chaining in Java?

Answer: Constructor chaining is the **process of calling one constructor from another constructor** with respect to current object.

Constructor chaining can be done in two ways:

**Within same class:** It can be done using this() keyword for constructors in same class

**From base class:** by using super() keyword to call constructor from the base class.

Constructor chaining occurs through inheritance.

### 5.1.40 The difference between Serial and Parallel Garbage Collector?

Answer: The serial collector uses a single thread to perform all garbage collection work, which is more efficient because of no communication overhead between threads. It is best suited to single processor machines.

Parallel collector performs minor collections in parallel, which can significantly reduce garbage collection. It is intended for applications with medium-size to large sized data sets that are run on multiprocessor and multi threaded hardware.

### 5.1.41 What is JIT stands for?

Answer: The Just-In-Time (JIT) compiler is a component of the Java™ Runtime Environment that improves the performance of Java applications at run time.The JIT compiler helps improve the performance of Java programs by compiling bytecodes into native machine code at run time.The JIT compiler is enabled by default, and is activated when a Java method is called.

### 5.1.42 Explain Java Heap space and Garbage collection?

Answer:

Heap space: At run-time the Java instances are stored in the heap memory area. When an object is not referenced anymore it becomes eligible for removal from heap memory.During garbage collection process, those objects are removed from heap memory and the space is reclaimed.

**Heap memory has three major areas**

1. Young Generation

a. Eden Space (any instance enters the runtime memory area through eden)

b. S0 Survivor Space (older instances moved from eden to S0)

c. S1 Survivor Space (older instances moved from S0 to S1)

2. Old Generation (instances promoted from S1 to tenured)

3. Permanent Generation (contains meta information like class, method detail)

Garbage Collection:**Java garbage collection is an automatic process to manage the runtime memory used by programs**.Garbage Collection of JAVA application is done by Garbage collector.

A garbage collector performs automatic dynamic memory management through the following operations:

1. It allocates from and gives back memory to the operating system.

2. Allocate memory to the application as it requests it.

3. Determines which parts of that memory is still in use by the application.

4. Reclaims the unused memory for reuse by the application.

An object is considered garbage and its memory can be reused by the JVM when it can no longer be reached from any reference of any other live object in the running program.

### 5.1.43 Can you guarantee the garbage collection process?

Answer:No it is not guaranteed process. Reason is as follows~

Being an automatic process, programmers need not initiate the garbage collection process explicitly in the code. System.gc() and Runtime.gc() are hooks to request the JVM to initiate the garbage collection process.

Though this request mechanism provides an opportunity for the programmer to initiate the process but the onus is on the JVM. It can choose to reject the request and so it is not guaranteed that these calls will do the garbage collection. This decision is taken by the JVM based on the eden space availability in heap memory.

The JVM specification leaves this choice to the implementation and so these details are implementation specific.

**Undoubtedly we know that the garbage collection process cannot be forced.**

### 5.1.44 What is the difference between stack and heap in Java?

Heap and Stack are part of JVM but they are used for different purposes.

Differences are as follows:

1. Size: Heap space in Java is much bigger than the Stack memory.

2. Resizing: JVM allows to resize both heap and stack, but need to use different JVM flags for that.

3. Usage: Heap memory is used to store objects, no matter where u created that object. Stack memory is used to

store primitive variables and reference variables, method frames and call stack.

4. Visibility: Heap memory is shared by all threads hence it is also known as the main memory but stack memory

is local to threads and local variable created there was not visible to others.

5. Order: Heap is a large memory area where objects can be created and stored in any order but Stack memory

is structured as Stack data structure i.e. LIFO where method calls are stored as last in first out order.

6. Error Type: When heap and stack memory are filled different error are being generated.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Reference: http://www.techbeamers.com/java-interview-questions/

### 5.1.45 Q: What is JRE and why is it required?

JRE stands for "Java Runtime Environment" which you usually download as a Java software. **The JRE comprises of the Java Virtual Machine, Java platform classes, and supporting libraries**. The JRE is the runtime component of Java software and is all you need to run any Java application.

### 5.1.46 Q: What is JDK and why is it required?

**The JDK is a superset of the JRE and includes everything that the JRE contains.** Additionally, it comes with the compilers and debuggers tools required for developing Java applications.

### 5.1.47 Q: What is JVM and why is it required?

JVM stands for The Java Virtual machine. **It translates and executes the Java bytecode**. It's the entity which transform Java to become a "portable language" (i.e. write once, run anywhere). Though, each platform has its implementation of JVM like the Windows, Linux, MacOS, etc. have a distinct version of JVM to run bytecode.

### 5.1.48 Q: Distinguish between the Path and Classpath?

**The <Path> and <Classpath> are OS level environment variables**. Path defines the location where the system can look up for the executables (.exe) files, and classpath specifies the location of the Java class files(Jar).

Jar:

A JAR (Java ARchive) is a [package](https://en.wikipedia.org/wiki/Package_format) [file format](https://en.wikipedia.org/wiki/File_format) typically used to aggregate many [**Java class files**](https://en.wikipedia.org/wiki/Java_class_file)and **associated** [**metadata**](https://en.wikipedia.org/wiki/Metadata)and **resources (text, images, etc.)** into one file for distribution.

JAR files are [archive files](https://en.wikipedia.org/wiki/Archive_file) that include a Java-specific [manifest file](https://en.wikipedia.org/wiki/Manifest_file). They are built on the [ZIP format](https://en.wikipedia.org/wiki/ZIP_(file_format)) and typically have a .jar [file extension](https://en.wikipedia.org/wiki/File_extension).

<https://www.youtube.com/watch?v=MupLMFz0O1s>

### 5.1.49 Q: Distinguish between a constructor and method?

A constructor gets automatically invoked to create an object whereas the method gets called explicitly.

Constructor shall have same name as class name but method can have any name.

Constructor must not have return type but method must have return type.

**Constructor** compiler provide default constructor where as **method** compiler does't provide

### 5.1.50 Q: Is it permissible for a constructor to have a different name than its class name in Java?

No, constructors in Java should have the same name as their classes. If the name is different, then it would behave like a standard method.

### 5.1.51 Q: Is there any difference between an argument and a parameter?

While defining methods, you pass variables which you refer as parameters. And when you call these methods and pass values for the variables then they are phrased as arguments.

### 5.1.52 Q: How your program would behave if you declare the main method as private?

It would get compiled correctly but will throw the error "Main method not public." at runtime.

### 5.1.53 Q: What if an application get multiple classes having main() methods?

It's certainly possible to have multiple main methods in different classes. When you start the application, you've to provide the startup class name for execution. The JVM then looks up for the main method only in the class whose name you've supplied. Hence, you won't observe any conflict with the multiple classes having the <main()> definition.

### 5.1.54 Q: What difference you see between pass by reference and pass by value in Java?

Pass by reference indicates, passing the address itself rather than passing the value. Pass by value means is giving a copy of the value.

### 5.1.55 Q: What do you understand by Byte Code?

Java compiler generates bytecode for all the Java code and converts into class files. The bytecode is platform independent and needs the platform-specific JVM for the execution.

### 5.1.56 Q: What do you make of each keyword in public static void main(String args[])?

* Public- <main(..)> is the entry point method which the JVM calls when a program is run. So it's mandatory for it to get accessible from the Java environment. Hence, the access specifier has to be public.
* Static- JVM must be capable of calling this method w/o creating an instance of the class. So the method has to be declared as static.
* Void- <main()> doesn't return anything so it's return type must be void.
* The argument string represents the argument type passed from the console, and the <args> is an array of strings specified at the command line.

### 5.1.57 Q: How do compare the final, finally and finalize keywords?

* final– It's used to declare a constant.
  + Variables defined in an interface are implicitly final.
  + You can't extend a final class.
* finally– It makes you handle exceptions.
  + It's a keyword used for exception handling. The code under the <finally> block gets executed apparently.
* finalize– It helps in garbage collection.
  + The <finalize()> method is used just before an object is destroyed and garbage collected.

### 5.1.58 Q: Can you compile a Java class successfully without having the main method?

Yes, we can compile, but it won't run. The <main> method works as the startup function for a Java class, and the JVM calls it for the program execution.

### 5.1.59 Q: What do you make of System, out and <println> in the function System.out.println()?

### 5.1.60 Second Explanation

* *System* is a class defined in the *java.lang* package.(final class)
* out is an instance of *PrintStream* , which is a public and static member of the class *System*.
* As all instances of *PrintStream* class have a public method *println()*, hence we can invoke the same on out as well. We can assume *System.out*  represents the standard Output Stream .

System Class: public final class **System**  
extends [Object](https://docs.oracle.com/javase/9/docs/api/java/lang/Object.html)

The System class contains several useful class fields and methods.**System Class cannot be instantiated.**Among the facilities provided by the System class are standard input, standard output, and error output streams; access to externally defined properties and environment variables; a means of loading files and libraries; and a utility method for quickly copying a portion of an array.

### 5.1.61 Q: What do you understand by the explicit casting?

It's a process which instructs the compiler about transforming the object into a different type.

**e.g.** *long no = 99999;*

*int new\_no = (int) no; // Explicit casting*

### 5.1.62 Q: Would a Java program compile/run if we use <static public void> instead of <public static void>?

Yes, the program will compile and run as usual.

### 5.1.63 Q: How would you prove that an array is not null but is empty?

Call the <Print array.length>. It will print 0. That suggests that the array is empty. If it would've been null then, it would've thrown a <*NullPointerException*> on calling the <Print array.length>.

### 5.1.64 Q: What do you understand of Garbage Collection and how to call it explicitly?

If the object is no longer belong to any variable, Java automatically reclaims the memory. This process is known as garbage collection. You can use the **<System.gc()> method to call it explicitly**.

### 5.1.65 Q: How an unreachable object become reachable again, is it at all possible?

Yes, an unreachable object may get to reachable state. It can happen if the object <finalize> method gets called during the garbage collection, and there you have set an object making a reference to it. This situation would cause the garbage collection to skip and make the object reachable again.

Reference: <http://www.techbeamers.com/java-collection-interview-questions-developers/#q1>

### 5.1.66 Java Collection Interview Questions And Answers.

#### 5.1.66.1 What Is The Use Of Java Collections Framework?

Java Collections Framework (JCF) represents a **set of interfaces and classes** which provide efficient ways to store and handle data in a Java application. It includes several classes that support operations like searching, sorting, insertion, manipulation, and deletion.

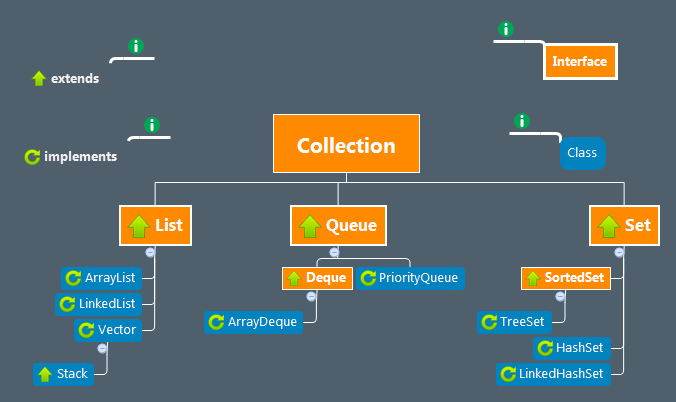
#### 5.1.66.2 What Are The Main Benefits Of Java Collections Framework?

There are primarily four qualities that make Java Collections Framework extremely useful for the programmers.

* **Performance –** Collection classes are highly efficient data structures to improve speed and accuracy.
* **Reusability –** Collection classes can intermix with other types to promote reusability.
* **Maintainability –** Supports consistency and interoperability in the implementation and hence makes the code easy to maintain.
* **Extensibility –** Allows customization of primitive collection types as per developer’s requirements.

#### 5.1.66.3 What Is The Structure Of Java Collections Framework?

Below is the diagram picturing the different collection classes and interfaces.



**Java Collection Framework Hierarchy.**

#### 5.1.66.4 What Is The Difference Between Collection And Collections Framework?

* **Collection –** A collection is an object container that can hold multiple elements and functions as a single unit.
* **Collections Framework –** Collections framework provides a centralized system for managing and representing collections.

#### 5.1.66.5 What Is The Difference Between Collection And Collections?

* The Collection is an interface while Collections is a Java class.
* Both the Collection and Collections are part of **<java.util>** package and are the main components of Java Collections Framework.

#### 5.1.66.6 What Is The Root Interface In The Collection Hierarchy?

It is **Collection interface which is the root interface in the collection hierarchy.** However, a few may disagree and will say that the Collection interface extends the Iterable interface. So Iterable must be the root interface.

But you can make the counter-argument by saying that the Iterable interface is a part of the **<java.lang>**package, not of the **<java.util>** package.

Also, the Oracle’s Java collections documentation states that Collection interface is a member of the Java Collections framework which belongs to the **<java.util>** package.

#### 5.1.66.7 What Are The Primary Interfaces In The Java Collections Framework?

**Java Collections Framework comprises of five basic interfaces.**

* **Collection Interface –** Most of the collections in Java inherit from the Collection interface. It is the core of the collections framework and stays at the root of Collection’s hierarchy.
* **Set Interface –** It is a mathematical interpretation of **Set which doesn’t allow duplicate entries.** Also, a set doesn’t define any order for the elements and hence doesn’t support the index-based search.
* **Queue Interface –** This interface follows the **principals of a queue and stores elements in the same order as they enter.** Operations like addition will take place at the rear and removal from the front.
* **List Interface –** A list is an extended form of an array. **Lists are ordered and can also contain duplicate elements.** Not only it allows the index-based search, but insertion can also be done independently of the position.
* **Map Interface –** Map is a **two-dimensional data structure which stores data in the form of Key-Value pair.** The map is **essentially a set and can’t have duplicate elements.** In a map, the value represents an element whereas the key is the hash code of the same element.

#### public class **Collections** extends [Object](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)

This class consists exclusively of static methods that operate on or return collections. It contains polymorphic algorithms that operate on collections, "wrappers", which return a new collection backed by a specified collection, and a few other odds and ends.

The methods of this class all throw a NullPointerException if the collections or class objects provided to them are null.

#### 5.1.66.9 What is HashSet?

**HashSet extends AbstractSet and implements the Set interface.** It creates a collection that uses a hash table for storage.

A hash table stores information by using a mechanism called **hashing**. In hashing, the informational content of a key is used to determine a **unique value,** **called its hash code.**

The hash code is then used as the index at which the data associated with the key is stored. The transformation of the key into its **hash code is performed automatically.**

**Declaration**

**Set<String> hashSet = new HashSet<String>();**

#### 5.1.66.10 Which Are The Collection Classes That Are Synchronized Or Thread-Safe?

Following Collection classes are thread-safe and support synchronization. You can use them safely in a multi-threaded Java application.

* **Hashtable –** A Hashtable is **a collection class which stores lists** as array elements also known as buckets. It associates keys to values. None of the key or value can be null. And the keys are objects which are hashed to compute the hash code. The resultant works as the index to store the value within the table.
* **Vector –** A Vector class serves as a dynamic array. It behaves like ArrayList with two main differences. The first is that a Vector is synchronized and secondly, it brings in some additional methods which are not part of the Collections Framework.

#### 5.1.66.11 What Is The Primary Difference Between List And Set?

#### 5.1.66.12 List Vs Set.

* **Set holds unique elements** whereas **List can have duplicate elements**.
* The **Set is an unordered** collection whereas **List is the ordered** one.
* The List preserves the order of the object addition.

#### 5.1.66.13 What Is The Principal Difference Between Map And Set?

Map object use **unique keys to hold** values whereas **Set contains unique values.**

#### 5.1.66.14 Which Are The Classes That Implement List And Set Interface?

Following classes implement the List interface.

* ArrayList,
* Vector, and
* LinkedList.

Following classes implement the Set interface.

* HashSet, and
* TreeSet

#### 5.1.66.15 What Is An Iterator In The Java Collections Framework?

The Iterator is **an interface**. You can locate it in the **<java.util>** package. **It provides methods to iterate over any Collection.**

#### 5.1.66.16 What Are The Principal Differences Between Iterator And Enumeration?

#### 5.1.66.17 Iterator Vs Enumeration.

* **Enumeration:**
  + Enumeration functions like a read-only interface as it only allows to traverse and retrieve the object.
  + Enumeration is doubly faster than the Iterator and uses very less memory.
  + Enumeration is very necessary and fulfills the primary requirements.
* **Iterator**
  + The Iterator is much safer than Enumeration as it always blocks other threads to alter the collection object.
  + Iterator gives the ability to remove elements using remove() method whereas Enumeration doesn’t.
  + Iterator method names are more logical to make their functionality clear.

#### 5.1.66.18 What Is The Design Pattern That Iterator Uses?

* It uses iterator design pattern. Iterator design pattern enables the ability to sail through the collection of objects by using a standard interface abstracting the underlying implementation.
* Enumeration is also an example of Iterator design pattern.

#### 5.1.66.19 What Are The Methods To Override For You To Use Any Object As A Key In HashMap?

To use an object as the key in HashMap, you can implement equals() and hashCode() methods.

#### 5.1.66.20 What Is The Main Difference Between Queue And Stack?

3.1.65.19 Queue Vs Stack.

* **The Queue** is a type of data structure which follows the FIFO (first in first out) mechanism.
  + An example of Queue in the real world is casting your votes in the electoral center.
* **The Stack** is a data structure which follows the LIFO (last in first out) mechanism.
  + A real-time example of Stack is when a person wears a bracelet set, the last bracelet worn is the first to get removed and the opening bracelet you would take out in the end.

#### 5.1.66.21 What Is The Best Way To Reverse A List In Collections?

The Collections class provides a reverse(List list) method which accepts the list as a parameter.

e.g. **Collections.reverse(listOfobject);**

#### 5.1.66.22 Which Approach Would You Choose To Convert The Array Of Strings Into The List?

There is one method in the Arrays class of java.util package as List() which takes the array as its parameter.

List<String> arrayList = new ArrayList<String>();  
arrayList.add("techbeamers")  
String [] strArr = arrayList.toArray(new String[arrayList.size()]);

#### 5.1.66.23 Iterator Doesn’t Have A Method To Get Next Element. Can You Explain Why?

It would be easy to introduce a method on top of current Iterator interface, but its use will be rare. Hence, it doesn’t make sense to add it to the interface that everyone has to implement.

#### 5.1.66.24 What Is The Difference Between The Iterator And ListIterator?

* You can use Iterator to traverse the collections like Set and List whereas ListIterator you can utilize with Lists only.
* Using Iterator, you can cross in the forward direction only whereas ListIterator supports to traverse in both the direction.
* ListIterator inherits from Iterator interface and brings add-on functionalities like the ones listed below.
  + Adding an element,
  + Replacing an element, and
  + Retrieve index position for the previous and next elements.

#### 5.1.66.25 What Is Iterator’s Fail-Fast Property?

* Iterator fail-fast property verifies for any change that occurs in the structure of the underlying collection whenever you try to get the next element.
* If it detects any change, it throws the ConcurrentModificationException.
* All the Iterator implementations in the Collection classes are fail-fast by design except the concurrent collection classes like ConcurrentHashMap and CopyOnWriteArrayList.

#### 5.1.66.26 What Is The Difference Between Fail-Fast And Fail-Safe?

3.1.65.26 Fail-Fast Vs Fail-Safe.

* Fail-fast is nothing but the mechanism to support the instant reporting of any failure and whenever a problem occurs fail fast system fails.
* In Java, a fail-fast iterator may throw concurrent modification exception whereas a fail-safe iterator won’t. The exception can occur due to one of the two reasons as given below.
  + When one of the threads is iterating through the collection while the other one is trying to modify it.
  + Or after invoking the remove() method if you attempt to change the collection object.

#### 5.1.66.27 What Are The Different Ways To Sort A List Of Objects?

Following methods can be applied to perform sorting.

* To sort the array of objects, you can use Arrays.sort() method.
* When you need to sort a collection of the objects, then use Collections.sort() method.

#### 5.1.66.28 How Does HashMap Work In Java?

HashMap follows the concept of hashing. It employs put(key, value) and get(key) methods to manage (storing and retrieving) the objects from the HashMap. When we call the put() method with a key and value object, the HashMap mechanism triggers the hashCode() method on the key object and returns the hash value. Then, the hashing function locates a bucket location using the hash value where it can save the key-value pair in the form of a nested class called Entry (Map.Entry).

When you need to fetch a value from the HashMap, you call the get() method with the desired key as the parameter. The key object produces the same hash value which leads to same bucket location where the object was stored in the put() method.

#### 5.1.66.29 What Is The Difference Between HashMap And Hashtable?

* Synchronization or Thread Safe
* Null keys and null values
* Iterating the values
* Default Capacity

#### 5.1.66.30 Why Map Interface Doesn’t Extend Collection Interface?

* The Set is a unordered collection and does not allow duplicate elements.
* The List is ordered collection allows duplicate elements whereas the Map is a key-value pair.
* It is viewed as a set of keys and collection of values.
* A Map is a collection of key-value pairs so by design they separated from collection interface.

#### 5.1.66.31 What Is The Use Of Properties Class?

The Properties is a subclass of Hashtable. You can use it to manage the list of values using the keys, both the key and value are of String type.

#### 5.1.66.32 How To Remove The Redundant Elements From An ArrayList Object?

You can apply the following to cut off the redundant element from the ArrayList.

* Offload all of the ArrayList elements to a LinkedHashSet. It helps to eliminate the duplicates and keeps the insertion order intact.
* Emptying the ArrayList
* Copy all elements of LinkedHashSet (non-duplicates) to an ArrayList.

You can refer the below example for help.

|  |
| --- |
| import java.util.ArrayList; import java.util.List; import java.util.LinkedHashSet; public class DeleteDuplicates {   public static void main(String[] args) {    */\* Creating an ArrayList of Strings and adding  \* all elements to it  \*/*  List<String> arrlst = new ArrayList<String>();  arrlst.add("tech");  arrlst.add("beamers");  arrlst.add("java");  arrlst.add("python");  arrlst.add("selenium");  arrlst.add("shell");    *// Displaying the ArrayList elements*   System.out.println("Before:");  System.out.println("ArrayList contains: "+arrlst);    *// Creating a LinkedHashSet*  LinkedHashSet<String> linkhs = new LinkedHashSet<String>();    */\* Adding ArrayList elements to the LinkedHashSet  \* in order to remove the duplicate elements and   \* to preserve the insertion order.  \*/*  linkhs.addAll(arrlst);    *// Removing the ArrayList elements*  arrlst.clear();    *// Adding the LinkedHashSet elements to the ArrayList*  arrlst.addAll(linkhs);    *// Displaying ArrayList elements*  System.out.println("After:");  System.out.println("ArrayList contains: "+arrlst);  } } |

## 5.2 Java Programming Problems for Testers

<http://www.techbeamers.com/java-coding-questions-software-testers/#q4>

<http://www.abodeqa.com/2015/07/25/commonly-asked-java-programs-in-selenium-interview/>

What's New in JDK 8  
  
Java Platform, Standard Edition 8 is a major feature release. This document summarizes features and enhancements in Java SE 8 and in JDK 8, Oracle's implementation of Java SE 8. Click the component name for a more detailed description of the enhancements for that component.  
  
Java Programming Language  
  
Lambda Expressions, a new language feature, has been introduced in this release. They enable you to treat functionality as a method argument, or code as data. Lambda expressions let you express instances of single-method interfaces (referred to as functional interfaces) more compactly.  
  
Method references provide easy-to-read lambda expressions for methods that already have a name.  
  
Default methods enable new functionality to be added to the interfaces of libraries and ensure binary compatibility with code written for older versions of those interfaces.  
  
Repeating Annotations provide the ability to apply the same annotation type more than once to the same declaration or type use.  
  
Type Annotations provide the ability to apply an annotation anywhere a type is used, not just on a declaration. Used with a pluggable type system, this feature enables improved type checking of your code.  
  
Improved type inference.  
  
Method parameter reflection.  
  
Collections  
  
Classes in the new java.util.stream package provide a Stream API to support functional-style operations on streams of elements. The Stream API is integrated into the Collections API, which enables bulk operations on collections, such as sequential or parallel map-reduce transformations.  
  
Performance Improvement for HashMaps with Key Collisions  
  
Compact Profiles contain predefined subsets of the Java SE platform and enable applications that do not require the entire Platform to be deployed and run on small devices.  
  
Security  
  
Client-side TLS 1.2 enabled by default  
  
New variant of AccessController.doPrivileged that enables code to assert a subset of its privileges, without preventing the full traversal of the stack to check for other permissions  
  
Stronger algorithms for password-based encryption  
  
SSL/TLS Server Name Indication (SNI) Extension support in JSSE Server  
  
Support for AEAD algorithms: The SunJCE provider is enhanced to support AES/GCM/NoPadding cipher implementation as well as GCM algorithm parameters. And the SunJSSE provider is enhanced to support AEAD mode based cipher suites. See Oracle Providers Documentation, JEP 115.  
  
KeyStore enhancements, including the new Domain KeyStore type java.security.DomainLoadStoreParameter, and the new command option -importpassword for the keytool utility  
  
SHA-224 Message Digests  
  
Enhanced Support for NSA Suite B Cryptography  
  
Better Support for High Entropy Random Number Generation  
  
New java.security.cert.PKIXRevocationChecker class for configuring revocation checking of X.509 certificates  
  
64-bit PKCS11 for Windows  
  
New rcache Types in Kerberos 5 Replay Caching  
  
Support for Kerberos 5 Protocol Transition and Constrained Delegation  
  
Kerberos 5 weak encryption types disabled by default  
  
Unbound SASL for the GSS-API/Kerberos 5 mechanism  
  
SASL service for multiple host names  
  
JNI bridge to native JGSS on Mac OS X  
  
Support for stronger strength ephemeral DH keys in the SunJSSE provider  
  
Support for server-side cipher suites preference customization in JSSE  
  
JavaFX  
  
The new Modena theme has been implemented in this release. For more information, see the blog at fxexperience.com.  
  
The new SwingNode class enables developers to embed Swing content into JavaFX applications. See the SwingNode javadoc and Embedding Swing Content in JavaFX Applications.  
  
The new UI Controls include the DatePicker and the TreeTableView controls.  
  
The javafx.print package provides the public classes for the JavaFX Printing API. See the javadoc for more information.  
  
The 3D Graphics features now include 3D shapes, camera, lights, subscene, material, picking, and antialiasing. The new Shape3D (Box, Cylinder, MeshView, and Sphere subclasses), SubScene, Material, PickResult, LightBase (AmbientLight and PointLight subclasses) , and SceneAntialiasing API classes have been added to the JavaFX 3D Graphics library. The Camera API class has also been updated in this release. See the corresponding class javadoc for javafx.scene.shape.Shape3D, javafx.scene.SubScene, javafx.scene.paint.Material, javafx.scene.input.PickResult, javafx.scene.SceneAntialiasing, and the Getting Started with JavaFX 3D Graphics document.  
  
The WebView class provides new features and improvements. Review Supported Features of HTML5 for more information about additional HTML5 features including Web Sockets, Web Workers, and Web Fonts.  
  
Enhanced text support including bi-directional text and complex text scripts such as Thai and Hindi in controls, and multi-line, multi-style text in text nodes.  
  
Support for Hi-DPI displays has been added in this release.  
  
The CSS Styleable\* classes became public API. See the javafx.css javadoc for more information.  
  
The new ScheduledService class allows to automatically restart the service.  
  
JavaFX is now available for ARM platforms. JDK for ARM includes the base, graphics and controls components of JavaFX.  
  
Tools  
  
The jjs command is provided to invoke the Nashorn engine.  
  
The java command launches JavaFX applications.  
  
The java man page has been reworked.  
  
The jdeps command-line tool is provided for analyzing class files.  
  
Java Management Extensions (JMX) provide remote access to diagnostic commands.  
  
The jarsigner tool has an option for requesting a signed time stamp from a Time Stamping Authority (TSA).  
  
Javac tool  
  
The -parameters option of the javac command can be used to store formal parameter names and enable the Reflection API to retrieve formal parameter names.  
  
The type rules for equality operators in the Java Language Specification (JLS) Section 15.21 are now correctly enforced by the javac command.  
  
The javac tool now has support for checking the content of javadoc comments for issues that could lead to various problems, such as invalid HTML or accessibility issues, in the files that are generated when javadoc is run. The feature is enabled by the new -Xdoclint option. For more details, see the output from running "javac -X". This feature is also available in the javadoc tool, and is enabled there by default.  
  
The javac tool now provides the ability to generate native headers, as needed. This removes the need to run the javah tool as a separate step in the build pipeline. The feature is enabled in javac by using the new -h option, which is used to specify a directory in which the header files should be written. Header files will be generated for any class which has either native methods, or constant fields annotated with a new annotation of type java.lang.annotation.Native.  
  
Javadoc tool  
  
The javadoc tool supports the new DocTree API that enables you to traverse Javadoc comments as abstract syntax trees.  
  
The javadoc tool supports the new Javadoc Access API that enables you to invoke the Javadoc tool directly from a Java application, without executing a new process. See the javadoc what's new page for more information.  
  
The javadoc tool now has support for checking the content of javadoc comments for issues that could lead to various problems, such as invalid HTML or accessibility issues, in the files that are generated when javadoc is run. The feature is enabled by default, and can also be controlled by the new -Xdoclint option. For more details, see the output from running "javadoc -X". This feature is also available in the javac tool, although it is not enabled by default there.  
  
Internationalization  
  
Unicode Enhancements, including support for Unicode 6.2.0  
  
Adoption of Unicode CLDR Data and the java.locale.providers System Property  
  
New Calendar and Locale APIs  
  
Ability to Install a Custom Resource Bundle as an Extension  
  
Deployment  
  
For sandbox applets and Java Web Start applications, URLPermission is now used to allow connections back to the server from which they were started. SocketPermission is no longer granted.  
  
The Permissions attribute is required in the JAR file manifest of the main JAR file at all security levels.  
  
Date-Time Package - a new set of packages that provide a comprehensive date-time model.  
  
Scripting  
  
The Rhino javascript engine has been replaced with the Nashorn Javascript Engine  
  
Pack200  
  
Pack200 Support for Constant Pool Entries and New Bytecodes Introduced by JSR 292  
  
JDK8 support for class files changes specified by JSR-292, JSR-308 and JSR-335  
  
IO and NIO  
  
New SelectorProvider implementation for Solaris based on the Solaris event port mechanism. To use, run with the system property java.nio.channels.spi.Selector set to the value sun.nio.ch.EventPortSelectorProvider.  
  
Decrease in the size of the <JDK\_HOME>/jre/lib/charsets.jar file  
  
Performance improvement for the java.lang.String(byte[], \*) constructor and the java.lang.String.getBytes() method.  
  
java.lang and java.util Packages  
  
Parallel Array Sorting  
  
Standard Encoding and Decoding Base64  
  
Unsigned Arithmetic Support  
  
JDBC  
  
The JDBC-ODBC Bridge has been removed.  
  
JDBC 4.2 introduces new features.  
  
Java DB  
  
JDK 8 includes Java DB 10.10.  
  
Networking  
  
The class java.net.URLPermission has been added.  
  
In the class java.net.HttpURLConnection, if a security manager is installed, calls that request to open a connection require permission.  
  
Concurrency  
  
Classes and interfaces have been added to the java.util.concurrent package.  
  
Methods have been added to the java.util.concurrent.ConcurrentHashMap class to support aggregate operations based on the newly added streams facility and lambda expressions.  
  
Classes have been added to the java.util.concurrent.atomic package to support scalable updatable variables.  
  
Methods have been added to the java.util.concurrent.ForkJoinPool class to support a common pool.  
  
The java.util.concurrent.locks.StampedLock class has been added to provide a capability-based lock with three modes for controlling read/write access.  
  
Java XML - JAXP  
  
HotSpot  
  
Hardware intrinsics were added to use Advanced Encryption Standard (AES). The UseAES and UseAESIntrinsics flags are available to enable the hardware-based AES intrinsics for Intel hardware. The hardware must be 2010 or newer Westmere hardware. For example, to enable hardware AES, use the following flags:  
  
-XX:+UseAES -XX:+UseAESIntrinsics  
To disable hardware AES use the following flags:  
  
-XX:-UseAES -XX:-UseAESIntrinsics  
Removal of PermGen.  
  
Default Methods in the Java Programming Language are supported by the byte code instructions for method invocation.  
  
Java Mission Control 5.3 Release Notes  
  
JDK 8 includes Java Mission Control 5.3.

# 6 Domain Knowledge

## 6.1 Ecommerce Domain

### 6.1.1 What is Web Portal and give some Examples?

Web portal is a business gateway ex: online shopping portals, http://www.naukri.com/ like job portals

### 6.1.2 What are the Types of Ecommerce Applications available in the IT Industry?

Business to Business (B2B)

Business to Customers (B2C)

Customers to Customers (C2C)

Customers to Business (C2B)

Etc...

### 6.1.3 What are two advantages of Electronic Commerce over Traditional Commerce?

Advantages of electronic commerce over traditional commerce are as follows:

1. Instant worldwide availability.

2. A streamlined buyer-to-seller relationship.

3. Reduced paperwork.

4. Reduced errors.

5. Time and overhead costs.

6. Reduced time to complete transactions.

7. Easier entry into new markets.

8. New business opportunities.

9. Improved market analysis.

10. Wider access to assistance and advice.

11. Improved product analysis.

12. The ability to streamline and automate purchasing.

### 6.1.4 How Testing is crucial in ECommerce?

Testing is crucial to e-commerce because e-commerce sites are both business critical and highly visible to their users; any failure can be immediately expensive in terms of lost revenue and even more expensive in the longer term if disaffected users seek alternative sites. Yet the time pressures in the e-commerce world militate against the thorough testing usually associated with business criticality, so a new approach is needed to enable testing to be integrated into the development process and to ensure that testing does not present a significant time burden.

### 6.1.5 How to test Payment Gateways in Web Portals?

Using dummy card ids, we can test Payment Gateways.

### 6.1.6 What is Online Shopping Application and give some examples?

Online Shopping Application provides business gateway between Product vendors and Customers.

Ex: Walmart, amazon etc...

### 6.1.7 7) What is the difference between business-to-business and business-to-consumer Electronic Commerce?

Difference between business-to-business and business-to-consumer e-commerce:

Business-to-business electronic commerce is conducted between two separate businesses, such as a large company needing office supplies and an office supply company. Business-to-consumer electronic commerce is between one individual and a company selling goods or services.

### 6.1.8 What are the possible drawbacks of Electronic Commerce?

Increased vulnerability to fraud; difficulty protecting intellectual property; risks to confidentiality; problems over taxation; customs requirements; regulations; credit card fraud; security; trust problems, and constant availability.

### 6.1.9 What are the payment models for Electronic Commerce?

The e-cash model, the check model, and the credit model.

### 6.1.10 How does Web marketing create a more personalized approach than radio or television advertising?

By allowing users to select the ads they would like to pursue; in radio and television advertising, the viewer or listener is more of a passive recipient of the information.

### 6.1.11 How to set Up an ECommerce System?

E-commerce Web sites are not easy to set up. With a plethora of e-commerce solutions in the market, entrepreneurs have to make a few key decisions:

• The entrepreneur has to decide on the initial amount of investment required for an e-commerce Web site, as well as the volume of business of an e-commerce Web site over the Internet. Investment factors and business objectives dictate the type of software, database, or other applications that are required to set up the e-commerce Web site.

• There are specific elements involved in an e-commerce system. These elements range from domain name for the site to the merchant account for e-commerce transactions. Each of these elements requires a certain amount of scrutiny before setting up an e-commerce Web site.

• Before launching the e-commerce Web site on the Internet, it requires rigorous testing. Some of the important and common types of testing include security testing, software and hardware reliability, and compatibility between all the elements of the system.

### 6.1.12 What is Software Reliability in ECommerce?

E-commerce requires software that performs critical tasks, such as creating storefront and a shopping cart, collecting customer data, and providing the payment gateway. This software needs to function correctly.

Testing assures the organization of the quality and integrity of the e-commerce solution.

### 6.1.13 What is System Assurance in Ecommerce?

The main purpose of system assurance is to deliver a quality product. Conformance to requirements increases the organization’s confidence in the system.

**An e-commerce system deals with three parties: the bank, the transaction clearinghouse, and the customer.** The interdependency of these three parties makes the process of buying and selling over the Internet more critical than in real life. If the faith of any of these parties dwindles in the e-commerce site, the entrepreneurs can lose a lot of money, as well as their reputation.

For example, in the case of a faulty e-commerce system, the credit card of the customer may be billed immediately for the complete order, when only a partial order has been filled. Testing must assure that partial order fulfillment and billing are done correctly.

## 6.2 E-commerce Testing - Why Is Testing Important In The E-Commerce Application?

**Introduction**

E-commerce applications are online business applications where the users sell and buy products. **E-commerce** is a software and business process which allows businesses to work through internet that is digitally. The E-commerce applications have various business processes like online fund transfers, marketing, inventory management, supply chain management. Developing an e-commerce website, we need to find out designs and features which are customer friendly. It is important that these features are properly working and doesn’t have any bugs in it, to give the customer a nice and enjoyable experience thus; e-commerce website testing is an essential part of the development of e-commerce website.

The e- commerce website testing requires knowledge of the web based testing techniques. The *E-commerce* website testers test their websites continuously. Testing is done in different browsers, different platforms, various combinations are tested and also testing occurs on devices like mobiles and iPad.

## 6.3 Why Do We Test E-Commerce Websites?

The e-commerce applications are very critical for businesses.The e-commerce application testing ensures that all the pages of the website are thoroughly tested all the e-commerce transactions are secure and validated and the application can be now given to end users to use. The applications promise to provide the fast delivery and at very competitive prices and are in huge demand always. The e-commerce websites are the future of the world, growing at the fast rate and also massively, it requires a lot of investment, thus it requires and extensive testing. The e-commerce applications have been in since a long time now and there has been a lot of failure in this field, if we look in the history. **These failures have lead to heavy losses which could have been avoided with better testing techniques**. To check the usability of the application, how user friendly it is, make the application defect free are the reason we perform testing on e-commerce application. Maintain Quality Assurance standards are also vital part of a software development, so to see if our application comply to the quality assurance standards we perform testing

### 6.3.1 What To Test In E-Commerce Websites?

****

Some common things which should be tested in e-commerce applications are:

### 6.3.2 Testing the workflow of the application

The testing of the complete workflow would include

1. Login and Register to the website
2. Search Functionality of the Product
3. Applying filters on the Product range
4. Sorting feature in the website
5. Add/Remove to the shopping cart
6. Post Reviews on the Products
7. Check Out process
8. Payment Gateway and Payment Processing
9. Generation of order number and invoice

### 6.3.3 Functionality of the application

Basically, an e-commerce application has structure like:

### 6.3.4 Main pages

1. Home Page
2. Product Page
3. Special Offers
4. Sitemap Pages like information , about us

### 6.3.5 Product Type /category pages

1. The product page have options like size, color , type
2. Sorting feature is present to sort size, price
3. There is a feature to add to cart or add to wish list.

### 6.3.6 Product Detail Page

1. Product Details page has product title, description
2. Images of the Product
3. Relates products
4. Information like features, compare to a related product
5. Add to cart feature

### 6.3.7 Shopping Cart

1. View the list of products
2. Remove the product from the list
3. Select delivery option, cash on delivery, card payment
4. Pay now

**We need to understand the e-commerce website functionality to test the website.** The features mentioned above are the common features of all e-commerce sites, but it may also be customized on business needs.

### 6.3.8 Payment Gateway of the application

Test the checkout process and payment process taking in consideration the following things:

1. Final Amount to be paid: Check that the price is correct state and federal tax, shipping charges, any discount codes all are applied and the final amount is right. Test with making changes in the final list of products, applying different discount codes, selecting different area to see the change in shipping charges.
2. Payments: Check if the payment is processed correctly by using all kinds of payment methods. Methods like Debit Card, Credit Card, Net Banking, Paypal. Check using demo card numbers and dummy accounts. Also Check if the orders are cancelled, the payment id sent back
3. Check the generation of invoice and the emails that are sent when payment is made.
4. Also ensure that the refund process and email, receipt of refund are all working fine.

### 6.3.9 Security and Vulnerability Issues

Perform security testing in the application to make sure that the website is not prone to any security issues. Techniques like SQL Injections, Ethical hacks on various pages like Log in, Register. Also check that the payment gateway is secured.

### 6.3.10 Browser Compatibility

It is important for the e-commerce applications to work on all types browsers, since if the customer uses one particular type of browser on which application does not work then he might not be able to use the website. The old browsers are now obsolete and new browsers are used generally. The latest browsers used include, IE7 and above, Good Chrome, Firefox and Safari.

### 6.3.11 Compatibility on Mobile Devices

If the website is build for use on mobile phone and website then it should be tested on various devices as well. Devices like Apple iPhone, Apple IPad, Apple IPad Mini, Android Phones, Windows Phone.

### 6.3.12 Performance and SEO

The e-commerce application should load fast so that the users get the best experience. Use performance testing tools to test the performance of the website.

The website should be searchable by the user easily, so it should have good Search engine optimization. It can be achieved by using unique title tags, Structure of URLs, Sitemaps.

### 6.3.13 SEO:

The process of maximizing the number of visitors to a particular website by ensuring that the site appears high on the list of results returned by a search engine.

### 6.3.14 Other common things which should be tested in the e-commerce applications:

1. Content of the application
2. Formatting of the pages
3. Adding content /Deleting Content
4. Making changes in the settings
5. Removing and adding links
6. Making changes in shipping settings
7. Web Standards
8. Accessibility to the website.
9. Cookies
10. Analytics.
11. Social Buttons

Reference: <http://www.softwaretestinghelp.com/ecommerce-testing/>

**Card Details Storage** – does the site store customer’s credit card details? If so are they securely stored? Is it [PCI compliant](https://www.pcisecuritystandards.org/)?

**What is PCI?**

**A:** The Payment Card Industry Data Security Standard (PCI DSS) is a set of security standards designed to ensure that ALL companies that accept, process, store or transmit credit card information maintain a secure environment.

<https://www.pcicomplianceguide.org/faq/#1>

**Challenges of E-commerce Testing**

* Compliance with security guidelines to safeguard customer data and identity
* Compliance with accessibility standards to support multi-lingual markets and business regions
* End to end testing and test management for large e-commerce transformation programs
* Scalability and reliability of applications

# 7 General Interview Questions

## 7.1 Tell me about yourself

About your current project

## 7.2 Tell me about yourself.

Good Morning!

Thanks for giving me this golden opportunity to introduce myself before this panel.

I am Md Jahidul Islam. I have been working in software testing industry more than six years, and experienced in Manual as well as Automated testing of web and client/ server

Applications.

As a detail oriented QA Tester, I successfully developed test plans, test cases, test scripts, traceability matrices and attended zillion meetings with the Business Analysts, Project Managers, Business Managers, Software Developers and QA Leads. As far as different types of testing, I have successfully performed and lead others to perform Smoke Testing, Functional Testing, Backend Testing, Black Box Testing, Integration Testing, Regression Testing, Load Testing and Stress Testing, constantly documenting and sharing knowledge about test projects with team members.

As a software test engineer, I was involved in all stages of software development life

cycle in multiple projects of different company that I previously worked with and have

deeper understanding of the development processes such as waterfall, iterative, and

agile methodology.

I am expert in working with

* Project Management and Defect Tracking Tools **HP ALM (12.50)** &  **JIRA(7.2 Release on August 2016)**
* Automation Tool Selenium WebDrive**r (3.7.1, Dec 15, 2017)**
* Appium(**1.6.3 December 12, 2016**)
* Git as version control and source code management **( v2.14.2 released on sep 22 17 and latest version v2.16.1 released on 15 days ago**)
* Windows, Unix/Linux
* Eclipse(**4.7 (Oxygen) / 28 June 2017**),

IntelliJ IDEA(**Version 2017.2.6 & Released on Nov 13, 2017**)

* Build tool Maven(**24 October 2017, 3.5.2** )
* Testing Framework: TestNG(**6.13.1**)/JUnit(Version **4.12**, **Dec, 2014**)
* Programming Knowledge Java(**JDK 1.8/9.0**)
* **Cloud Testing Environment**: Sauce Lab, BrowserStack
* **AWS: EC2**, S3 Buckets
* **Web-Services:** REST API Testing with REST Assured
* **Database:** Oracle, SQL Server, MySQL, PL/SQL
* **Jenkins 2.95 (2017-12-14)**

I worked with windows operating systems mainly but I have working knowledge of UNIX and LINUX.

Recently I have automated a major portion of our manual test suite using Selenium and by that we were able to save lots of effort spent on repetitive work.

Recently I was appreciated for suggesting design changes in our application UI.

This is pretty much what I have been doing as a QA Tester in the past years.

## 7.3 Did you get any compliments from your previous employers? What were those situations?

Answer: Yes. I did. There were many occasions where I had compliments. For example, I was testing an application going a little bit off my test cases. After I finished executing my test cases, I always think in a way what a real user would possibly click in various parts of the application. So I was just clicking back and forth and at one specific scenario, the application simply broke and displayed an error message. That scenario was not in the test cases. The manager really appreciated me and thanked for finding this kind of critical defect.

Answer: Yes. I did.There were many occasions where I had compliments. For example, I was testing an application going a little bit off my test cases. After I finished executing my test cases, I always think in a way what a real user would possibly click in various parts of the application. So I was just clicking back and forth and at one specific scenario, the application simply broke and displayed an error message. That scenario was not in the test cases. The manager really appreciated me and thanked for finding this kind of critical defect.

# 8 AWS

Top 15 AWS Interview Questions

## 8.1 1) Explain what is AWS?

AWS stands for Amazon Web Service; it is a collection of remote computing services also

known as cloud computing platform. This new realm of cloud computing is also known as IaaS or Infrastructure as a Service.

## 8.2 2) Mention what are the key components of AWS?

The key components of AWS are

**Route 53**: A DNS web service

**Simple E-mail Service:** It allows sending e-mail using RESTFUL API call or via regular

SMTP

**Identity and Access Management:** It provides enhanced security and identity

management for your AWS account

**Simple Storage Device or (S3):** It is a storage device and the most widely used AWS

service

**Elastic Compute Cloud (EC2):** It provides on-demand computing resources for hosting

applications. It is very useful in case of unpredictable workloads

**Elastic Block Store (EBS):** It provides persistent storage volumes that attach to EC2 to

allow you to persist data past the lifespan of a single EC2

**CloudWatch:** To monitor AWS resources, It allows administrators to view and collect

key Also, one can set a notification alarm in case of trouble.

## 8.3 3) Explain what is S3?

S3 stands for **Simple Storage Service**. You can use S3 interface to store and retrieve any

amount of data, at any time and from anywhere on the web. For S3, the payment model is “pay as you go”.

## 8.4 4) Explain what is AMI?

**AMI stands for Amazon Machine Image**. It’s a template that provides the information (an

operating system, an application server and applications) required to launch an instance, which ,is a copy of the AMI running as a virtual server in the cloud. You can launch instances from as many different AMIs as you need.

## 8.5 5) Mention what is the relation between an instance and AMI?

**From a single AMI, you can launch multiple types of instances.** An instance type defines the hardware of the host computer used for your instance. Each instance type provides different compute and memory capabilities. Once you launch an instance, it looks like a traditional host, and we can interact with it as we would with any computer.

### 8.5.1 What does an AMI include?

An AMI includes the following things

A template for the root volume for the instance

Launch permissions decide which AWS accounts can avail the AMI to launch instances

A block device mapping that determines the volumes to attach to the instance when it is

launched

## 8.6 7) How can you send request to Amazon S3?

Amazon S3 is a REST service, you can send request by using the REST API or the AWS SDK wrapper libraries that wrap the underlying Amazon S3 REST API.

## 8.7 8) Mention what is the difference between Amazon S3 and EC2?

The difference between EC2 and Amazon S3 is that

EC2 S3

EC2 It is a cloud web service used for hosting your application and

S3 is a data storage system where any amount of data can be stored.

EC2 It is like a huge computer machine which can run either Linux or Windows.

S3 It has a REST interface and uses secure HMACSHA1 authentication keys

handle application like PHP,Python, Apache or any databases

## 8.8 9) How many buckets can you create in AWS by default?

By default, you can create upto 100 buckets in each of your AWS accounts.

## 8.9 10) Explain can you vertically scale an Amazon instance? How?

Yes, you can vertically scale on Amazon instance. For that

Spin up a new larger instance than the one you are currently running

Pause that instance and detach the root webs volume from the server and discard

Then stop your live instance and detach its root volume. Note the unique device ID and attach that root volume to your new server And start it again

## 8.10 11) Explain what is T2 instances?

T2 instances are designed **to provide moderate baseline performance and the capability** to burst to higher performance as required by workload.

## 8.11 12) In VPC with private and public subnets, database servers should ideally be launched into which subnet?

With private and public subnets in VPC, database servers should ideally launch into **private**

**subnets.**

## 8.12 13) Mention what are the security best practices for Amazon EC2?

For secure Amazon EC2 best practices, follow the following steps

* Use AWS identity and access management to control access to your AWS resources
* Restrict access by allowing only trusted hosts or networks to access ports on your instance
* Review the rules in your security groups regularly
* Only open up permissions that your require
* Disable password-based login, for instance, launched from your AMI

## 8.13 14) Explain how the buffer is used in Amazon web services?

The buffer is used to make the system more robust to manage traffic or load by synchronizing different component. Usually, components receive and process the requests in an unbalanced way, With the help of buffer, the components will be balanced and will work at the same speed to provide faster services.

## 8.14 15) While connecting to your instance what are the possible connection issues one might face?

The possible connection errors one might encounter while connecting instances are

* Connection timed out
* User key not recognized by the server
* Host key not found, permission denied
* Unprotected private key file
* Server refused our key or No supported authentication method available
* Error using MindTerm on Safari Browser
* Error using Mac OS X RDP Client

# 9 Jenkins

## 9.1 Software Configuration Management Administrator

**1) Mention what is Jenkins?**

Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

**2) Explain what is continuous integration?**

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

**3) What is the requirement for using Jenkins?**

To use Jenkins you require

* A source code repository which is accessible, for instance, a Git repository
* A working build script, e.g., a Maven script, checked into the repository

**4) Mention what are the advantages of Jenkins?**

Advantage of Jenkins include

* At integration stage, build failures are cached
* For each code commit changes an automatic build report notification generates
* To notify developers about build report success or failure, it is integrated with LDAP mail server
* Achieves continuous integration agile development and test driven development
* With simple steps, maven release project is automated
* Easy tracking of bugs at early stage in development environment than production

**5) Explain how you can move or copy Jenkins from one server to another?**

* Slide a job from one installation of Jenkins to another by copying the related job directory
* Make a copy of an already existing job by making clone of a job directory by a different name
* Renaming an existing job by renaming a directory.

**6) Mention what are the commands you can use to start Jenkins manually?**

To start Jenkins manually, you can use either of the following

* (Jenkins\_url)/restart: Forces a restart without waiting for builds to complete
* (Jenkin\_url)/safeRestart: Allows all running builds to complete

**7) Mention some of the useful plugins in Jenkin?**

Some of the important plugins in Jenkin includes

* Maven 2 project
* Amazon EC2
* HTML publisher
* Copy artifact
* Join
* Green Balls

**8) Explain how you can deploy a custom build of a core plugin?**

To deploy a custom field of a core plugin, you have to do following things

* Stop Jenkins
* Copy the custom HPI to $Jenkins\_Home/plugins
* Delete the previously expanded plugin directory
* Make an empty file called <plugin>.hpi.pinned
* Start Jenkins

**9) Explain how can create a backup and copy files in Jenkins?**

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

**10) Explain how you can clone a Git repository via Jenkins?**

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. For that, you have to switch into your job directory and execute the “git config” command.

**11) Explain how you can set up Jenkins job?**

To create a project that is handled via jobs in Jenkins. Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins. The next page enables you to configure your job.

**12) Mention what are the two components Jenkins is mainly integrated with?**

Jenkin is mainly integrated with two components

* Version Control system like GIT, SVN
* And build tools like Apache Maven.
* JDK

<https://codingcompiler.com/jenkins-interview-questions-answers/>

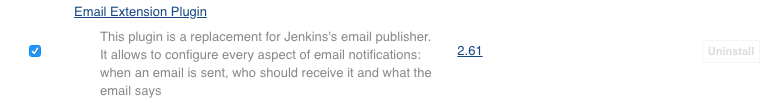
## 9.2 Send Email at every build with Jenkins(By Mahmudul Hasan)

## 9.3 Environment

* Email Extension Plugin
* Jenkins
* Maven
* git
* Oracle JDK

## 9.4 Install Email Extension Plugin

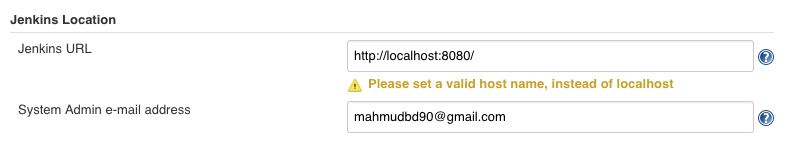
Install Email Extension Plugin at plug-in install page of Jenkins



## 9.5 Configure System

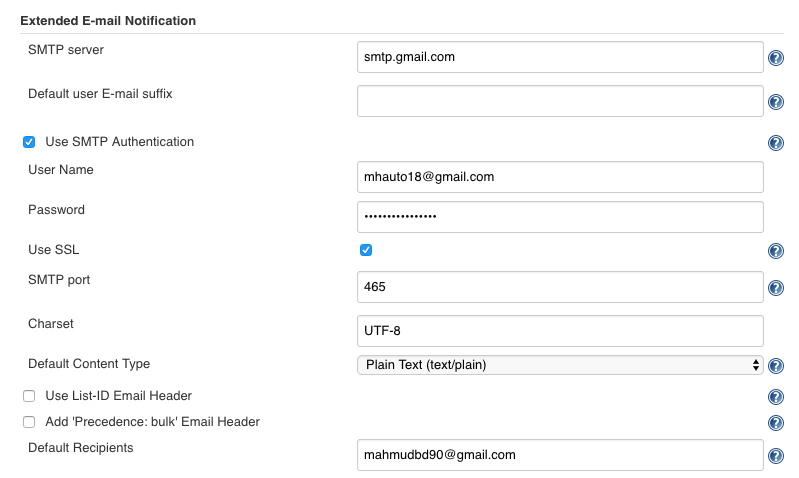
### 9.5.1 “Jenkins Location” section

Enter valid email address to “System Admin email address”



### 9.5.2 “Extended Email Notification” section

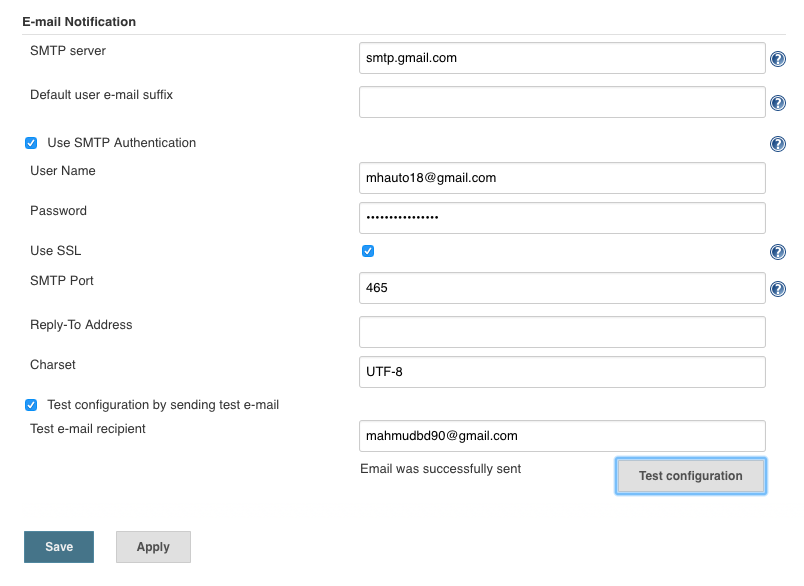
* Enter your SMTP server name to “SMTP server”
* Click “Advanced”
* Click “Use SMTP Authentication”
* Enter required informations
* Enter your email address to “Default Recipients”



### 9.5.3 “Email Notification” section

* Enter your SMTP server name to “SMTP server”
* Click “Advanced”
* Click “Use SMTP Authentication”
* Enter required informations
* Check “Test configuration by sending test email”
* Click “Test configuration” to send test email
* Click “Save” in the bottom of the page

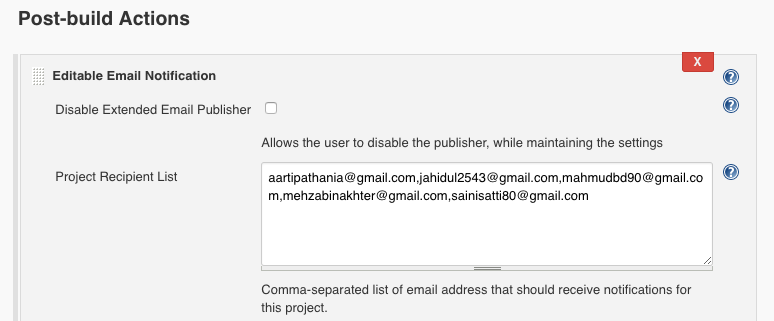
Get an apps pwd from google: <https://www.lifewire.com/get-a-password-to-access-gmail-by-pop-imap-2-1171882>

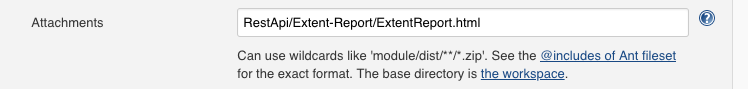


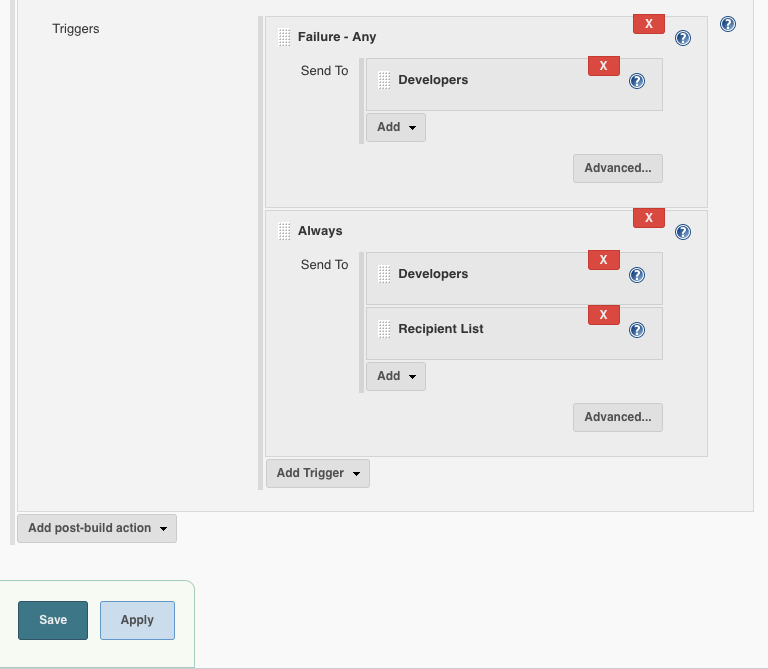
## 9.6 Configure a project to send email at every build

1. Click “Add post-build action”
2. Click “Editable Email Notification”
3. Click “Advanced Settings…”
4. Click “Add Trigger”
5. Click “Always”
6. Save









## 9.7 Test-run

1. Click “Build Now”
2. Check Console output and received email

## 9.8 References

Choose your configuration option

<https://support.google.com/a/answer/176600?hl=en>

Extent Report

<http://extentreports.com/docs/versions/2/java/#user-examples>

## 9.9 How to set up Jenkins for auto trigger whenever there is a event (git push) in github project?

**Genkins Side**

**create a jenkins job**.

write project name and descriptions,

check **<Github project>** and provide Github **<project url>**

In **<Source Code Management>** provide git repositry url(clone url) and appropriate branch

provide credentials by clicking <Add>

---

In the **<Build Triggers>**

Check **<GitHub hook trigger for GITScm polling>**

----

In the **<Build> se**ction

Invoke top level Maven targets

<Apply> and <Save>

----------

Go to **<Manage Jenkins>>>> <Configure System>**

**Get WebHook Url**

-------

GitHub Side

Go to git hub project<sttings>>><Integrations & Servics><Add Jenkins (GitHub plugin)>

<provide WebHook Url><add service><Test service// on top right side><>

>>Using the Jenkins GitHub Plugin you can automatically trigger build jobs when

pushes are made to GitHub.<<

Go to <WebHook>>>>WebHooks allow external services to be notified when

certain events(as selected in WebHook section, specially event is "push event")

happen. When the specified events happen, we’ll send a POST request to each of the URLs you provide.<<

<Payload url>This is webhook url>>> where GitHub will send POST request which will contain

information of new event (push event in github)>> also can specify which data format Jenkins

like to receive (JSON, x-www-form-urlencoded, etc) POST request.

----

Once Jenkins receives this request with the event(push) information it triggers the respective

Jenkins job.

------------------

**In Simple**

Create a jenkins freestyle job and select github project and git as Source Code Management

select "GitHub hook trigger for GITScm polling" in Build Trigger Sections, select rest items

Get WebHook url for respective project and got GitHub in Integrations & Services add services

Jenkins for GitHub plugins and then got WebHook and provide WebHook Url and select event git push. Whenever any push event happens in git project GitHub will send POST request in the respective WebHook url which contains information of events. Once Jenkins receives this Request Payloads it triggers jenkins job....

--------------------------

# 10 Git

**1) What is GIT?**

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

2) What is a repository in GIT?

**A repository contains a directory named .git**, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

**3) What is the command you can use to write a commit message?**

e command that is used to write a commit message is “git commit –a”. The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

**4) What is the difference between GIT and SVN?**

The difference between GIT and SVN is

a) Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.

b) **GIT does not support ‘commits’ across multiple branches or tags.**  Subversion allows the creation of folders at any location in the repository layout.

c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

**5) What are the advantages of using GIT?**

a) Data redundancy and replication

b) High availability

c) Only one.git directory per repository

d) Superior disk utilization and network performance

e) Collaboration friendly

f) Any sort of projects can use GIT

**6) What language is used in GIT?**

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

**7) What is the function of ‘GIT PUSH’ in GIT?**

‘GIT PUSH’ updates remote refs along with associated objects.

**8) Why GIT better than Subversion?**

GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes. Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

**9) What is “Staging Area” or “Index” in GIT?**

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**10) What is GIT stash?**

**GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.** So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

|  |
| --- |
| $ git stash list stash@{0}: WIP on master: 049d078 added the index file stash@{1}: WIP on master: c264051 Revert "added file\_size" stash@{2}: WIP on master: 21d80a5 added number to log *//Use this to apply git stash* git stash apply stash@{2} $ git stash apply # On branch master # Changes not staged for commit: # (use "git add <file>..." to update what will be committed) # # modified: index.html # modified: lib/simplegit.rb |

**11) What is GIT stash drop?**

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command.  **It will remove the last added stash item by default**, and it can also remove a specific item if you include as an argument.

**12) How will you know in GIT if a branch has been already merged into master?**

Git branch—merged lists the branches that have been merged into the current branch

Git branch—-no merged lists the branches that have not been merged

**13) What is the function of git clone?**

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, ‘cloning’ is the most common way used by programmers.

**14) What is the function of ‘git config’?**

The ‘git config’ command is a convenient way to set configuration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be defined through this command.

**15) What does commit object contain?**

a) A set of files, representing the state of a project at a given point of time

b) Reference to parent commit objects

c) An SHAI name, a 40 character string that uniquely identifies the commit object.

**16) How can you create a repository in Git?**

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

**17) What is ‘head’ in git and how many heads can be created in a repository?**

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”. A repository can contain any number of heads.

**18) What is the purpose of branching in GIT?**

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

**19) What is the common branching pattern in GIT?**

The common way of creating branch in GIT is to maintain one as “Main“

branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

**20) How can you bring a new feature in the main branch?**

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

**21) What is a ‘conflict’ in git?**

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

**22) How can conflict in git resolved?**

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

**23) To delete a branch what is the command that is used?**

Once your development branch is merged into the main branch, you don’t need

development branch. To delete a branch use, the command “git branch –d [head]”.

**24) What is another option for merging in git?**

“Rebasing” is an alternative to merging in git.

**25) What is the syntax for “Rebasing” in Git?**

The syntax used for rebase is “git rebase [new-commit] “

**26) What is the difference between ‘git remote’ and ‘git clone’?**

‘git remote add’ just creates an entry in your git config that specifies a name for a particular URL. While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

**27) What is GIT version control?**

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version. Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

**28) Mention some of the best graphical GIT client for LINUX?**

Some of the best GIT client for LINUX is

a) Git Cola

b) Git-g

c) Smart git

d) Giggle

e) Git GUI

f) qGit

**29) What is Subgit? Why to use Subgit?**

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration. Subgit is a solution for a company -wide migration from SVN to Git that is:

a) It is much better than git-svn

b) No requirement to change the infrastructure that is already placed

c) Allows to use all git and all sub-version features

d) Provides genuine stress –free migration experience.

**30) What is the function of ‘git diff ’ in git?**

‘git diff ’ shows the changes between commits, commit and working tree etc.

**31) What is ‘git status’ is used for?**

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

**32) What is the difference between the ‘git diff ’and ‘git status’?**

‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

**33) What is the function of ‘git checkout’ in git?**

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

**34) What is the function of ‘git rm’?**

To remove the file from the staging area and also off your disk ‘git rm’ is used.

**35) What is the function of ‘git stash apply’?**

When you want to continue working where you have left your work, ‘**git stash apply’ command is used to bring back** the saved changes onto the working directory.

**36) What is the use of ‘git log’?**

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

**37) What is ‘git add’ is used for?**

‘git add’ adds file changes in your existing directory to your index.

**38) What is the function of ‘git reset’?**

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

**39) What is git Is-tree?**

‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

**40) How git instaweb is used?**

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

**41) What does ‘hooks’ consist of in git?**

This directory consists of Shell scripts which are activated after running the corresponding Git commands. For example, git will try to execute the post-commit script after you run a commit.

**42) Explain what is commit message?**

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

**43) How can you fix a broken commit?**

To fix any broken commit, you will use the command “git commit—amend”. By running this command, you can fix the broken commit message in the editor.

**44) Why is it advisable to create an additional commit rather than amending an existing commit?**

There are couple of reason

a) The amend operation will destroy the state that was previously saved in a commit. If it’s just the commit message being changed then that’s not an issue. But if the contents are being amended then chances of eliminating something important remains more.

b) Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

**45) What is ‘bare repository’ in GIT?**

To co-ordinate with the distributed development and developers team, especially when you are working on a project from multiple computers ‘Bare Repository’ is used. A bare repository comprises of a version history of your code.

**46) Name a few Git repository hosting services**

* Pikacode
* Visual Studio Online
* GitHub
* GitEnterprise
* SourceForge.net

# 

# 11 Various Article

**Automated Software Testing - A Perspective**

Note from the author: My perspective on most things is that the 'glass is half full' rather than half empty. This attitude carries over to the advice I suggest on automated [software testing](http://www.softwaretestingmagazine.com/) as well. I should point out, however, there is an increasing awareness from others experienced in this field, as well as from my own experience, that many efforts in test automation do not live up to expectations. A lot of effort goes into developing and maintaining test automation, and even once it's built you may or may not recoup your investment. The successes I've seen have mostly been on focused areas of the application where it made sense to automate, rather than complete automation efforts. Also, skilled people were involved in these efforts and they were allowed the time to do it right.

Test automation can add a lot of complexity and cost to a test team's effort, but it can also provide some valuable assistance if its done by the right people, with the right environment and done where it makes sense to do so. I hope by sharing some pointers that I feel are important that you'll find some value that translates into saved time, money and less frustration in your efforts to implement test automation back on the job.

**Key Points**

I’ve listed the ‘key points’ up front instead of waiting until the end. The rest of the article will add detail to some of these key points.

* First, it's important to define the purpose of taking on a test automation effort. There are several categories of testing tools each with its own purpose. Identifying what you want to automate and where in the testing life cycle will be the first step in developing a test automation strategy. Just wishing that everything should be tested faster is not a practical strategy. You need to be specific.
* Developing a test automation strategy is very important in mapping out what's to be automated, how it's going to be done, how the scripts will be maintained and what the expected costs and benefits will be. Just like every testing effort should have a testing strategy, or test plan, so should there be a 'plan' built for test automation.
* Many of the testing 'tools' provided by vendors are very sophisticated and use existing or proprietary coding 'languages'. The effort of automating an existing manual testing effort is no different than a programmer using a coding language to write programs to automate any other manual process. Treat the entire process of automating testing as you would any other software development effort. This includes defining what should be automated, (the requirements phase), designing test automation, writing the scripts, testing the scripts, etc. The scripts need to be maintained over the life of the product just as any program would require maintenance. Other components of software development, such as configuration management also apply
* The effort of test automation is an investment. More time and resources are needed up front in order to obtain the benefits later on. Sure, some scripts can be created which will provide immediate payoff, but these opportunities are usually small in number relative to the effort of automating most test cases. What this implies is that there usually is not a positive payoff for automating the current release of the application. The benefit comes from running these automated tests every subsequent release. Therefore, ensuring that the scripts can be easily maintained becomes very important.
* Since test automation really is another software development effort, it's important that those performing the work have the correct skill sets. A good tester does not necessarily make a good test automator. In fact, the job requirements are quite different. Good testers are still necessary to identify and write test cases for what needs to be tested. A test automator, on the other hand, takes these test cases and writes code to automate the process of executing those tests. From what I've seen, the best test automation efforts have been lead by developers who have put their energies into test automation. That's not to say that testers can't learn to be test automators and be successful, it's just that those two roles are different and the skill sets are different.

**Points**

Here are some other important points to consider:

When strategizing for test automation, plan to achieve small successes and grow. It's better to incur a small investment and see what the effort really takes before going gung ho and trying to automate the whole regression suite. This also gives those doing the work the opportunity to try things, make mistakes and design even better approaches.

Many software development efforts are underestimated, sometimes grossly underestimated. This applies to test automation as well, especially if the effort is not looked upon as software development. Test automation is not something that can be done on the side and care should be taken when estimating the amount of effort involved. Again, by starting small and growing, estimating the work can be gauged.

When people think of testing tools, many first think of the 'capture/playback' variety where the application is tested at the end during system test. There are several types of testing tools which can be applied at various points of code integration. Test automation can be applied at each of the levels of testing including unit testing, one or more layers of integration testing, and system testing (another form of integration). The sooner tests can be executed after the code is written, before too much code integration has occurred, the more likely bugs will not be carried forward. When strategizing for test automation, consider automating these tests as early as possible as well as later in the testing life cycle.

Related to this last point is the idea that testers and software developers need to work as a team to make effective test automation work. I don't believe testing independence is lost when testers and developers work together, but there can be some excellent advantages that I'll later point out.

Testing tools, as sophisticated as they have become, are still dependent upon consistency in the test environment. This should be quite obvious, but having a dedicated test environment is absolutely necessary. If testers don't have control of their test environment and test data, the required setup for tests may not meet the requirements of those tests. When manual testing is done testers may sometimes 'work around' test setup issues. Automated test scripts are less flexible and require specific setup scenarios, thereby needing more control.

Test automation is not the only answer to delivering quality software. In fact, test automation in many cases is a last gasp effort in an attempt to find problems after they've been made instead of eliminating the problems as they are being created. Test automation is not a substitute for walkthroughs, inspections, good project management, coding standards, good configuration management, etc. Most of these efforts produce higher pay back for the investment than does test automation. Testing will always need to be done and test automation can assist, but it should not be looked upon as the primary activity in producing better software.

The truth is that developers can produce code faster and faster with more complexity than ever before. Advancements in code generation tools and code reuse are making it difficult for testers to keep up with software development. Test automation, especially if applied only at the end of the testing cycle, will not be able to keep up with these advances. We must pull out all stops along the development life cycle to build in good quality software and test as early and often as possible with the assistance of test automation.

**Benefits**

To many people, the benefits of automation are pretty obvious. Tests can be run faster, they're consistent, and tests can be run over and over again with less overhead. As more automated tests are added to the test suite more tests can be run each time thereafter. Manual testing never goes away, but these efforts can now be focused on more rigorous tests.

There are some common 'perceived' benefits that I like to call 'bogus' benefits. Since test automation is an investment it is rare that the testing effort will take less time or resources in the current release. Sometimes there's the perception that automation is easier than testing manually. It actually makes the effort more complex since there's now another added software development effort. Automated testing does not replace good test planning, writing of test cases or much of the manual testing effort.

**Costs**

Costs of test automation include personnel to support test automation for the long term. As mentioned, there should be a dedicated test environment as well as the costs for the purchase, development and maintenance of tools. All of the efforts to support software development, such as planning, designing, configuration management, etc. apply to test automation as well.

**Common View**

Now that some of the basic points have been noted, I'd like to talk about the paradigm of testing automation. When people think of test automation, the 'capture/playback' paradigm is commonly perceived. The developers create the application software and turn it over to the testing group. The testers then busily use capture/playback functionality of the testing tool to quickly create test scripts. Capture/playback is used because it's easier than 'coding' scripts. These scripts are then used to test the application software.

There are some inherent problems with this paradigm. First, test automation is only applied at the final stage of testing when it is most expensive to go back and correct the problem. The testers don't get a chance to create scripts until the product is finished and turned over. At this point there is a tremendous pull on resources to just test the software and forgo the test automation effort. Just using capture/playback may be temporarily effective, but using capture/playback to create an entire suite will make the scripts hard to maintain as application modifications are made.

**Test and Automate Early**

From observations and experience, a different paradigm appears to be more effective. Just as you would want to test early and test often if you were testing manually, the same applies to test automation. The first level of testing is the unit testing performed by the developer. From my experience unit testing can be done well or not done well depending on the habits and personality of the developer. Inherently, developers like to develop, not write test cases. Here's where an opportunity for developers and testers to work together can begin to pay off. Testers can help document unit tests and developers can write utilities to begin to automate their unit tests. Assisting in documenting test cases will give a better measurement of unit tests executed. Much success of test automation comes from homegrown utilities. This is because they integrate so well with the application and there is support from the developer to maintain the utilities so that they work with the application. More effective and efficient unit testing, through the use of some automation, provides a significant bang for the buck in trying to find bugs in the testing life cycle. Static analyzers can also be used to identify which modules have the most code complexity and may require more testing.

**Work With Developers**

The same approach should be applied at each subsequent level of testing. Apply test automation where it makes sense to do so. Whether homegrown utilities are used or purchased testing tools, it's important that the development team work with the testing team to identify areas where test automation makes sense and to support the long-term use of test scripts.

Where GUI applications are involved the development team may decide to use custom controls to add functionality and make their applications easier to use. It's important to determine if the testing tools used can recognize and work with these custom controls. If the testing tools can't work with these controls, then test automation may not be possible for that part of the application. Similarly, if months and months of effort went into building test scripts and the development team decides to use new custom controls which don't work with existing test scripts, this change may completely invalidate all the effort that went into test automation. In either case, by identifying up front in the application design phase how application changes affect test automation, informed decisions can be made which affect application functionality, product quality and time to market. If test automation concerns aren't addressed early and test scripts cannot be run, there is a much higher risk of reduced product quality and increased time to market.

Working with developers also promotes building in 'testability' into the application code. By providing hooks into the application testing can sometimes be made more specific to any area of code. Also, some tests can be performed which otherwise could not be performed if these hooks were not built.

Besides test drivers and capture/playback tools, code coverage tools can help identify where there are holes in testing the code. Remember that code coverage may tell you if paths are being tested, but complete code coverage does not indicate that the application has been exhaustively tested. For example, it will not tell you what has been 'left out' of the application.

**Capture/Playback**

Here's just a note on capture/replay. People should not expect to install the testing tool, turn on the capture function and begin recording tests that will be used forever and ever. Capturing keystrokes and validating data captured within the script will make the script hard to maintain. Higher level scripts should be designed to be modular which has options to run several tests scripts. The lower level test scripts that actually perform tests also should be relatively small and modular so they can be shared and easily maintained. Data for input should not be hard coded into the script, but rather read from a file or spreadsheet and loop through the module for as many times as you wish to test with variations of data. The expected results should also reside in a file or spreadsheet and read in at the time of verification. This method considerably shortens the test script making it easier to maintain and possibly reuse by other test scripts. Bitmap comparisons should be used very sparingly. The problem with bitmap comparison is that if even one pixel changes in the application for the bitmap being compared, the image will compare as a mismatch even if you recognize it as a desirable change and not a bug. Again, the issue is maintainability of the test suite.

Capture/playback functionality can be useful in some ways. Even when creating small modular scripts it may be easier to first capture the test then go back and shorten and modify it for easier maintenance. If you wish to create scripts that will obviously provide immediate pay back, but you don't care if it's maintainable, then using capture/playback can be a very quick way to create the automated test. These scripts typically are thrown away and rebuilt later for long term use. The capture/playback functionality is also good to use during the design phase of a product if there's a prototype developed. During usability testing, which is an application design technique, users sit at the computer using a mock up of the actual application where they're able to use the interface, but the real functionality has not yet been built. By running the capture/playback tool in capture mode while the users are 'playing' with the application, recorded keystrokes and mouse movements can track where the users move on the system. Reading these captured scripts help the designers understand the level of difficulty in navigating through the application.

**Players**

Test automation is not just the responsibility of the testers. As noted, getting developers involved is important as well as getting the understanding and support of management. Since test automation is an investment, it's important that they understand the up front costs and expected benefits so that test automation stays around long enough to show the benefits. There is the tendency to 'give up' when results are not shown right away.

If the project is just beginning with test automation then having someone who can champion the test automation effort is important. This 'champion' should have skills in project management, software testing and software development (preferably a coding background). This 'champion' is responsible for being the project manager of the test automation effort. This person needs to interact well with both the testers and the application developers. Since this person may also be actively involved with writing scripts as well, good development skills are also desirable. This person should not be involved with the designing of test cases or manual testing other than to review other team member's work. Typically there is not enough time to both design test cases and design test automation. Nor is there time to build test scripts and run manual tests by the same person. Where the testing effort is large the distinction between these two roles apply to teams of automators and testers as well. Too many times test automators are borrowed to performed manual testing never to realize the benefits of test automation in the current or future releases of the application.

This is not to say that the role of testers is reduced. Test planning still needs to be done by a test lead, test cases still need to be designed and manual testing will still be performed. The added role for these testers is that they most likely will begin to run the automated test scripts. As they run these scripts and begin to work more closely with the test automation 'champion' or test automators, they too can begin to create scripts as the automated test suite matures.

Experience has shown that most bugs are not found by running automated tests. Most bugs are found in the process of creating the scripts, or the first time the code is tested. What test automation mostly buys you is the opportunity to not spend valuable man-hours re-testing code that has been tested before, but which has to be tested in any case because the risk is too high not to test it. The other benefit comes from the opportunity to spend these man-hours rigorously testing new code for the first time and identifying new bugs. Just as testing in general is not a guarantee, but a form of insurance, test automation is a method to have even more insurance.

**Some Nuts and Bolts**

When learning to use testing tools it's common to make mistakes. One way to mitigate these mistakes is to create scripts that will provide immediate pay back. That is, create scripts which won't take too much time to create yet will obviously save manual testing effort. These scripts will be immediately useful and it's all right if they're not intended to be part of the permanent test suite. Those creating the scripts will learn more about the tool's functionality and learn to design even better scripts. Not much is lost if these scripts are thrown away since some value has already been gained from them. As experience is gained with the testing tool, a long-term approach to test automation design can start to be developed.

Again, start off small when designing scripts. Identify the functional areas within the application being tested. Design at a higher level how each of these functional areas would be automated, then create a specific automated test design for one of the functional areas. That is, what approach will be used to create scripts using test cases as the basis for automating that function? If there are opportunities to use common scripting techniques with other testing modules, then identify these common approaches as potential standards would be useful in creating maintainable scripts.

Use a similar approach to design and create scripts for some of the other functional areas of the application. As more experience is gained from automation then designing and building scripts to test the integration of these functional areas would be the next step in building a larger and more useful testing suite.

Since the purpose of automating testing is to find bugs, validations should be made as tests are performed. At each validation point there is a possibility of error. Should the script find an error, logic should be built into it so that it can not only report the error it found but also route back to an appropriate point within the automated testing suite so that the automated testing can continue on. This is necessary if automated tests are to be run overnight successfully. This part of test automation is the 'error recovery process'. This is a significant effort since it has to be designed in for every validation point. It's best to design and create reusable error recovery modules that can be called from many validation points in many scripts. Related to this are the reports that get generated from running the tests. Most tools allow you to customize the reports to fit your reporting needs.

It's also important to write documented comments in the test scripts to help those who would maintain the test scripts. Write the scripts with the belief that someone else will be maintaining them.

In the automation test design or documented within the test scripts also identify any manual intervention which is necessary to set up the test environment or test data in order to run the scripts. Perhaps databases need to be loaded or data has to be reset.

**Test Data**

I know of three ways to have the test data populated so that the test environment is setup correctly to run automated tests. If complete control of the test environment is available to testers, then reloading preset databases can be a relatively quick way to load lots of data. One danger in having several preset databases is if a future release requires a reconstruction of data structures and the effort to convert the current data structures to the desired state is a large effort.

Another method of setting up the data is to create tests scripts which run and populate the database with the necessary data to be used in automated tests. This may take a little longer to populate, but there's less dependency on data structures. This method also allows more flexibility should other data change in the database. Even though I mention 'databases' specifically, the concepts apply to other types of data storage as well.

Other people with test automation experience have used 'randomly' generated data successfully to work with their test scripts. Personally, I have no experience using randomly generated data, but this is another option worth looking into if you're looking for other ways to work with data.

**Potential Risks**

Some common risks to the test automation effort include management and team members support fading after not seeing immediate results, especially when resources are needed to test the current release. Demanding schedules will put pressure on the test team, project management and funding management to do what it takes to get the latest release out. The reality is that the next release usually has the same constraints and you'll wish you had the automated testing in place.

If contractors are used to help build or champion the test automation effort because of their experience, there is the risk that much of the experience and skills will 'walk away' when the contractor leaves. If a contractor is used, ensure there is a plan to back fill this position since the loss of a resource most likely will affect the maintenance effort and new development of test scripts. It's also just as important that there is a comprehensive transfer of knowledge to those who will be creating and maintaining the scripts.

Since the most significant pay back for running automated tests come from future releases, consider how long the application being tested will remain in its current state. If a rewrite of the application is planned in the near future or if the interface is going to be overhauled, then it probably makes sense to only use test automation for immediate pay back. Again, here's where working with application designers and developers can make a difference, especially if internal changes are planned which may not appear to affect the testing team, but in reality can affect a large number of test scripts.

**Summary**

As mentioned earlier, most of the concepts identified here came from experiences and as also noted there are not a lot of facts to back up these ideas. The intent here wasn't to prove any particular technique worked, but, rather just to share methods that appear to be more successful. If nothing else, this information can be used to look at test automation from a little different perspective and assist in planning.

If you have experiences that are different than these that you've found successful, or if you've experienced hardships using some of these recommendations, I'd be grateful to hear from you. Many people, including myself, are interested in finding out what really works in creating higher quality software more quickly.

Reference: <http://www.methodsandtools.com/archive/archive.php?id=33>

## 11.1 Implementing Automated Software Testing - Continuously Track Progress and Adjust Accordingly

Thom Garrett, Innovative Defense Technologies, [www.IDTus.com](http://www.idtus.com/)

This is an excerpt from the book "Implementing Automated Software Testing," by Elfriede Dustin, Thom Garrett, Bernie Gauf, Copyright Addison Wesley, 2009

*"When you can measure what you are speaking about, and can express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind." - Lord Kelvin*

Most of us have worked on at least one project where the best-laid plans went awry and at times there wasn’t just one reason we could point to that caused the failure. People, schedules, processes, and budgets can all contribute. [1] Based on such experiences from the past, we have learned that as part of a successful automated testing (AST) program it is important that the right people with the applicable skills are hired, goals and strategies be defined and then implemented, and that steps be put in place to continuously track, measure, and adjust, as needed, against these goals and strategies. Here we’ll discuss the importance of tracking the AST program, to include various defect prevention techniques, such as peer reviews and other interchanges. We’ll then focus on the types of AST metrics to gather so that we can measure progress, gauge the effectiveness of our AST efforts, and help keep them keep on track and/or make adjustments, if necessary. Finally, we will discuss the importance of a root/cause analysis if a defect or issue is encountered.

Based on the outcome of these various efforts, adjustments can be made where necessary; e.g., the defects remaining to be fixed in a testing cycle can be assessed, schedules can be adjusted, and/or goals can be reduced. For example, if a feature is left with too many high-priority defects, a decision can be made to move the ship date, to ship the system as is (which generally isn’t wise, unless a quick patch process is in place), or to go live without that specific feature, if that is feasible.

Success is measured based on achieving the goals we set out to accomplish relative to the expectations of our stakeholders and customers.

**AST Program Tracking and Defect Prevention**

In our book "Implementing Automated Software Testing (IAST)" in we cover the importance of valid requirements and their assessment; in another we discuss the precautions to take when deciding what to automate; and yet another section we discussed in detail the importance of peer reviews. Here we’ll provide additional ideas that aid in defect prevention efforts, including technical interchanges and walk-throughs; internal inspections; examination of constraints and associated risks; risk mitigation strategies; safeguarding AST processes and environments via configuration management; and defining and tracking schedules, costs, action items, and issues/defects.

**Conduct Technical Interchanges and Walk-throughs**

Peer reviews, technical interchanges, and walk-throughs with the customer and the internal AST team represent evaluation techniques that should take place throughout the AST effort. These techniques can be applied to all AST deliverables—test requirements, test cases, AST design and code, and other software work products, such as test procedures and automated test scripts. They consist of a detailed artifact examination by a person or a group other than the author. These interchanges and walk-throughs are intended to detect defects, non-adherence to AST guidelines, test procedure issues, and other problems.

An example of a technical interchange meeting is an overview of test requirement documentation. When AST test requirements are defined in terms that are testable and correct, errors are prevented from entering the AST development pipeline that could eventually be reflected as defects in the deliverable. AST design component walk-throughs can be performed to ensure that the design is consistent with defined requirements - e.g., that it conforms to OA standards and applicable design methodology - and that errors are minimized.

Technical reviews and inspections have proven to be the most effective forms of preventing miscommunication, allowing for defect detection and removal.

**Conduct Internal Inspections**

In additionto customer technical interchanges and walk-throughs, internal inspections of deliverable work products should take place, before anything is even presented to the customer, to support the detection and removal of defects and process/practice omissions or deficiencies early in the AST development and test cycle; prevent the migration of defects to later phases; improve quality and productivity; and reduce cost, cycle time, and maintenance efforts.

**Examine Constraints and Associated Risks**

A careful examination of goals and constraints and associated risks should take place, leading to a systematic AST strategy and producing a predictable, higher-quality outcome and a high degree of success. Combining a careful examination of constraints together with defect detection technologies will yield the best results.

Any constraint and associated risk should be communicated to the customer and risk mitigation strategies developed as necessary.

**Implement Risk Mitigation Strategies**

Defined "light weight" processes allow for iterative, constant risk assessment and review without the dreaded overhead. If a risk is identified, appropriate mitigation strategies can be deployed. Require ongoing review of cost, schedules, processes, and implementation to ensure that potential problems do not go unnoticed until too late; instead, processes need to ensure that problems are addressed and corrected immediately. For example, how will you mitigate the risk if your "star" developer quits? There are numerous possible answers: Software development is a team effort and it is never a good practice to rely on one "star" developer. Hire qualified developers, so they can integrate as a team and each can be relied on in various ways based on their respective qualifications. One team member might have more experience than another, but neither should be irreplaceable, and the departure of one of them should not be detrimental to the project. Follow good hiring and software development practices (such as documenting and maintaining all AST-related artifacts) and put the right people on the project; we discuss the "how to" in our book "IAST." Additional risks could be missed deadlines or being over budget. Evaluate and determine risk mitigation techniques in case an identified risk comes to fruition.

**Safeguard the Integrity of the AST Process and Environments**

Experience shows that it is important to safeguard the integrity of the AST processes and environment. In IAST we discuss the importance of an isolated test environment and having it under configuration management. For example, you might want to test any new technology to be used as part of the AST effort in an isolated environment and validate that a tool, for example, performs to product specifications and marketing claims before it is used on any AUT or customer test environment. At one point we installed a tool on our Micron PC used for daily activities, only to have it blue-screen. It turned out that the tool we wanted to test wasn’t compatible with the Micron PC. To solve the problem, we actually had to upgrade the PC’s BIOS. An isolated test environment for these types of evaluation activities is vital.

The automator should also verify that any upgrades to a technology still run in the current environment. The previous version of the tool may have performed correctly and a new upgrade may perform fine in other environments, but the upgrade may adversely affect the team’s particular environment. We had an experience when a new tool upgrade wasn’t compatible with our e-mail software package any longer. It was a good thing we caught this issue, because otherwise an upgrade install would have rendered the tool useless, as we heavily relied on e-mail notification, for example, if a defect was generated.

Additionally, using a configuration management tool to baseline the test repository will help safeguard the integrity of the automated testing process. For example, all AST automation framework components, script files, test case and test procedure documentation, schedules, cost tracking, and other related AST artifacts need to be under configuration management. Using a configuration management tool ensures that the latest and most accurate version control and records of AST artifacts and products are maintained. For example, we are using the open-source tool Subversion in order to maintain AST product integrity; we evaluate the best products available to allow for the most efficient controls on an ongoing basis.

**Define, Communicate, and Track Schedules and Costs**

It is not good enough to base a schedule on a marketing-department-defined deadline. Instead, schedule and task durations need to be determined based on past historical performance and associated best estimates gathered from all stakeholders. Additionally, any schedule dependencies and critical path elements need to be considered up front and incorporated into the schedule. Project schedules need to be defined, continuously tracked, and communicated.

In order to meet any schedule—for example, if the program is under a tight deadline - only the AST tasks that can be successfully delivered in time are included in the schedule iteration. As described in IAST, during the AST Phase 1, test requirements are prioritized, which allows for prioritizing the most critical AST tasks to be completed as opposed to the less critical and lower-priority tasks, which can then be moved to later in the schedule, accordingly. Once the requirements are prioritized, an initial schedule is presented to the customer for approval and not before the System Under Test (SUT), AST requirements and associated level of effort are understood.

During the technical interchanges and walk-throughs, schedules are evaluated and presented on an ongoing basis to allow for continuous communication and monitoring. Potential schedule risks should be communicated well in advance and risk mitigation strategies explored and implemented, as needed; any schedule slips should be communicated to the customer immediately and adjustments made accordingly.

By closely tracking schedules and other required AST resources, we can also ensure that a cost tracking and control process is followed. Inspections, walk-throughs, and other status reporting allow for closely monitored cost control and tracking. Tracking cost and schedules and so forth allows for tracking of the project’s performance.

**Track Actions, Issues, and Defects**

A detailed procedure needs to be defined for tracking action items to completion. Templates should be used that describe all elements to be filled out for action item reports.

Additionally, a procedure needs to be in place that allows for tracking issues/defects to closure, known as a defect tracking lifecycle. See IAST for a sample defect tracking lifecycle used in the open-source defect tracking tool Bugzilla. Various defect tracking lifecycles exist; adapt one to your environment, tool, and project needs. Once defined, put measures in place to verify that the defect or action item lifecycle is adhered to.

If an issue or defect is uncovered, a root cause analysis should be conducted. See that section later on for more on root cause analysis.

**AST Metrics**

Metrics can aid in improving your organization’s automated testing process and tracking its status. Much has been said and written about the need for using metrics carefully and to not let metrics drive an effort, i.e. don’t measure for the sake of measuring. As with our recommended lightweight and adjustable process described in IAST, we recommend to use these metrics as an enhancement to the AST effort not to drive the AST effort. Our software test teams have successfully used the metrics and techniques discussed here. As the beginning quote implies, if you can measure something, then you have something you can quantify.

As time proceeds, software projects become more complex because of increased lines of code as a result of added features, bug fixes, etc. Also, tasks must be done in less time and with fewer people. Complexity over time has a tendency to decrease the test coverage and ultimately affect the quality of the product. Other factors involved over time are the overall cost of the product and the time in which to deliver the software. Carefully defined metrics can provide insight into the status of automated testing efforts.

When implemented properly, AST can help reverse the negative trend. As represented in Figure 1.1, automation efforts can provide a larger test coverage area and increase the overall quality of a product. The figure illustrates that the goal of automation is ultimately to reduce the time of testing and the cost of delivery, while increasing test coverage and quality. These benefits are typically realized over multiple test and project release cycles.

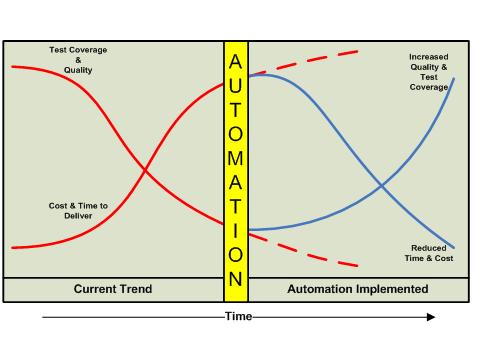


Figure 1.1 AST goal examples comparing current trend with automation implementation

Automated testing metrics can aid in making assessments as to whether coverage, progress and quality goals are being met. Before we discuss how these goals can be accomplished, we want to define metrics, automated testing metrics, and what makes a good automated test metric.

**What is a metric?** The basic definition of a *metric* is a standard of measurement. It can also be described as a system of related measures that facilitate the quantification of some particular characteristic. [2] **For our purposes, a metric can be seen as a** measure that can be used to display past and present performance and/or predict future performance.

**Metrics categories:** Most software testing metrics (including the ones presented here) fall into one of three categories:

* **Coverage:** meaningful parameters for measuring test scope and success.
* **Progress:** parameters that help identify test progress to be matched against success criteria. Progress metrics are collected iteratively over time. They can be used to graph the process itself (e.g., time to fix defects, time to test, etc.).
* **Quality:** meaningful measures of testing product quality. Usability, performance, scalability, overall customer satisfaction, and defects reported are a few examples.

**What are automated testing metrics?** Automated testing metrics are metrics used to measure the performance (past, present, and future) of the implemented automated testing process and related efforts and artifacts. Here we can also differentiate metrics related to unit test automation versus integration or system test automation. Automated testing metrics serve to enhance and complement general testing metrics, providing a measure of the AST coverage, progress, and quality, not replace them.

**What makes a good automated testing metric?** As with any metrics, automated testing metrics should have clearly defined goals for the automation effort. It serves no purpose to measure something for the sake of measuring. To be meaningful, a metric should relate to the performance of the effort.

Prior to defining the automated testing metrics, there are metrics-setting fundamentals you may want to review. Before measuring anything, set goals. What is it you are trying to accomplish? Goals are important; if you do not have them, what is it that you are measuring? It is also important to track and measure on an ongoing basis. Based on the metrics outcome, you can decide whether changes to deadlines, feature lists, process strategies, etc., need to be adjusted accordingly. As a step toward goal setting, questions may need to be asked about the current state of affairs. Decide what questions to ask to determine whether or not you are tracking toward the defined goals. For example:

* How many permutations of the test(s) selected do we run?
* How much time does it take to run all the tests?
* How is test coverage defined? Are we measuring test cases against requirements (generally during system testing), or are we measuring test cases against all possible paths taken through the units and components (generally used for unit testing)? In other words, are we looking at unit testing coverage, code coverage, or requirements coverage?
* How much time does it take to do data analysis? Are we better off automating that analysis? What would be involved in generating the automated analysis?
* How long does it take to build a scenario and required driver?
* How often do we run the test(s) selected?
* How many people do we require to run the test(s) selected?
* How much system and lab time is required to run the test(s) selected?

In essence, a good automated testing metric has the following characteristics:

* It is objective.
* It is measurable.
* It is meaningful.
* Data for it is easily gathered.
* It can help identify areas of test automation improvement.
* It is simple.

A few more words about metrics being simple: Albert Einstein once said, "Make everything as simple as possible, but not simpler." When applying this wisdom to AST and related metrics collection, you will see that

* Simplicity reduces errors.
* Simplicity is more effective.
* Simplicity is elegant.
* Simplicity brings focus.

It is important to generate a metric that calculates the value of automation, especially if this is the first time an automated testing approach has been used for a project. IAST discusses ROI measurement in detail and provides various worksheets that can serve as a baseline for calculating AST ROI. For example, there we mention that the test team will need to measure the time spent on developing and executing test scripts against the results that the scripts produce. If needed, the test team could justify the number of hours required to develop and execute AST by providing the number of defects found using this automation that would likely not have been revealed during a manual test effort. Specific details as to why the manual effort would not have found the defect can be provided; some possible reasons are that the automated test used additional test data not previously included in the manual effort, or the automated test used additional scenarios and path coverage previously not touched manually. Another way of putting this is that, for example, with manual testing you might have been able to test *x* number of test data combinations; with automated testing you are now able to test *x + y* test data combinations. Defects that were uncovered in the set of *y* combinations are the defects that manual testing may have never uncovered. Here you can also show the increase in testing coverage for future software releases.

Another way to quantify or measure automation benefits is to show that a specific automated test could hardly have been accomplished in a manual fashion. For example, say that during stress testing 1,000 virtual users execute a specific functionality and the system crashes. It would be very difficult to discover this problem manually, using 1,000 test engineers or possibly even extrapolation as it is still very commonly used today.

AST can also minimize the test effort, for example, by the use of an automated test tool for data entry or record setup. Consider the test effort associated with the system requirement that reads, "The system shall allow the addition of 10,000 new accounts." Imagine having to manually enter 10,000 accounts into a system in order to test this requirement! An automated test script can easily support this requirement by reading account information from a file through the use of a looping construct. The data file can easily be generated using a data generator. The effort to verify this system requirement using test automation requires far fewer man-hours than performing such a test using manual test methods. [3] The ROI metric that applies in this case measures the time required to manually set up the needed records versus the time required to set up the records using an automated tool.

What follows are additional metrics that can be used to help track progress of the AST program. Here we can differentiate between test case and progress metrics and defect and defect removal metrics.

**Percent Automatable or Automation Index**

As part of an AST effort, the project is either basing its automation on existing manual test procedures, or starting a new automation effort from scratch, some combination, or even just maintaining an AST effort. Whatever the case, a percent automatable metric or the automation index can be determined.

*Percent automatable* can be defined as the percentage of a set of given test cases that is automatable. This could be represented by the following equation:

ATC No. of test cases automatable  
 PA (%) = -------- = (----------------------------------- )  
 TC Total no. of test cases  
  
PA = Percent automatable  
ATC = Number of test cases automatable  
TC = Total number of test cases

When evaluating test cases to be developed, what is to be considered automatable and what is to be considered not automatable? Given enough ingenuity and resources, one can argue that almost anything can be automated. So where do you draw the line? Something that can be considered "not automatable," for example, could be an application area that is still under design, not very stable, and mostly in flux. In cases such as this, you should evaluate whether it makes sense to automate. See IAST for a detailed discussion of how to determine what to automate. There we discussed that we would evaluate, given the set of test cases, which ones would provide the biggest return on investment if automated. **Just because a test is automatable doesn’t necessary mean it should be automated.**

Prioritize your automation effort based on the outcome of this "what to automate" evaluation. Figure 1.2 shows how this metric can be used to summarize, for example, the percent automatable of various projects or components within a project and to set the automation goal. Once we know the percent automatable, we can use it as a baseline for measuring AST implementation progress.

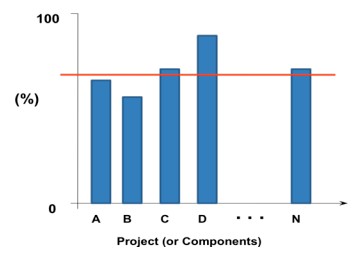


Figure 1.2 Example of percent automatable (automation index) per project (or component)

**Automation Progress**

*Automation progress* refers to the number of tests that have been automated as a percentage of all automatable test cases. Basically, how well are you doing against the goal of automated testing? The ultimate goal is to automate 100% of the "automatable" test cases. This can be accomplished in phases, so it is important to set a goal that states the deadlines for when a specific percentage of the ASTs should be automated. It is useful to track this metric during the various stages of automated testing development.

AA No. of test cases automated  
 AP (%) = -------- = (-------------------------------------- )  
 ATC No. of test cases automatable  
  
AP = Automation progress  
AA = Number of test cases automated  
ATC = Number of test cases automatable

The automation progress metric is a metric typically tracked over time. In the case of Figure 1.3, the time is weeks.

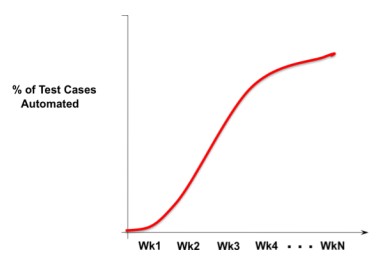


Figure 1.3 Test cases automated over time (weeks)

**Test Progress**

A common metric closely associated with the progress of automation, yet not exclusive to automation, is *test progress*. Test progress can simply be defined as the number of test cases (manual and automated) executed over time.

TC No. of test cases executed  
 TP = -------- = (-------------------------------------------- )  
 T Total number of test cases  
  
TP = Test progress  
TC = Number of test cases executed  
T = Total number of test cases

The purpose of this metric is to track test progress and can be used to show how testing is tracking against the overall project plan.

More detailed analysis is needed to determine test pass/fail, which can be captured in a more refined metric; i.e., we need to determine not only how many tests have been run over time and how many more there are to be run, but also how many of those test executions actually pass consistently without failure so that the test can actually be considered complete. In the test progress metric we can replace *No. of test cases executed* with *No. of test cases completed*,counting only those test cases that actually consistently pass.

**Percent of Automated Test Coverage**

Another AST metric we want to consider is *percent of automated test coverage*. This metric determines what percentage of test coverage the automated testing is actually achieving. Various degrees of test coverage can be achieved, depending on the project and defined goals. Also depending on the types of testing performed, unit test automation coverage could be measured against all identified units, or functional system test coverage can be measured against all requirements, and so forth. Together with manual test coverage, this metric measures the completeness of the test coverage and can measure how much automation is being executed relative to the total number of tests. However, it does not say anything about the quality of the automation. For example, 2,000 test cases executing the same or similar data paths may take a lot of time and effort to execute, but they do not equate to a larger percentage of test coverage. Test data techniques discussed in IAST need to be used to effectively derive the number of test data elements required to test the same or similar data path. Percent of automated test coverage does not indicate anything about the effectiveness of the testing taking place; it is a metric that measures its dimension.

AC Automation coverage  
 PTC (%) = ------- = (------------------------------- )  
 C Total coverage  
  
PTC = Percent of automated test coverage  
AC = Automation coverage  
C = Total coverage (i.e., requirements, units/components, or code coverage)

There is a wealth of material available regarding the sizing or coverage of systems. A useful resource is Stephen H. Kan’s book *Metrics and Models in Software Quality Engineering*. [4]

Figure 1.4 provides an example of test coverage for Project A versus Project B over various iterations. The dip in coverage for Project A might reveal that new functionality was delivered that hadn’t yet been tested, so that no coverage was provided for that area.

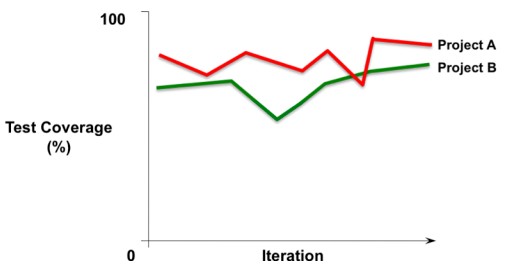


Figure 1.4 Test coverage per project over various iterations

**Defect Density**

Measuring defects is a discipline to be implemented regardless of whether the testing effort is automated or not. *Defect density* is another well-known metric that can be used for determining an area to automate. If a component requires a lot of retesting because the defect density is very high, it might lend itself perfectly to automated testing. Defect density is a measure of the total known defects divided by the size of the software entity being measured. For example, if there is a high defect density in a specific functionality, it is important to conduct a causal analysis. Is this functionality very complex, and therefore is it to be expected that the defect density would be high? Is there a problem with the design or implementation of the functionality? Were the wrong (or not enough) resources assigned to the functionality, because an inaccurate risk had been assigned to it and the complexity was not understood? It also could be inferred that the developer responsible for this specific functionality needs more training.

D No. of known defects  
 DD = ------- = ( ------------------------------- )  
 SS Size of software entity  
  
DD = Defect density  
D = Number of known defects  
SS = Size of software entity

We can’t necessary blame a high defect density on large software component size, i.e. while generally the thought is that a high defect density in a large component is more justifiable than in a small component, the small component could be much more complex than the large one. AST complexity is an important consideration when evaluating defect density.

Additionally, when evaluating defect density, the priority of the defect should be considered. For example, one application requirement may have as many as 50 low-priority defects and still pass because the acceptance criteria have been satisfied. Still another requirement may have only one open defect, but that defect prevents the acceptance criteria from being satisfied because it is a high priority. Higher-priority defects are generally weighted more heavily as part of this metric.

**Defect Trend Analysis**

Another useful testing metric in general is *defect trend analysis*. Defect trend analysis is calculated as:

D No. of known defects  
 DTA = ------- = (--------------------------------------- )  
 TPE No. of test procedures executed  
  
DTA = Defect trend analysis  
D = Number of known defects  
TPE = Number of test procedures executed over time

Defect trend analysis can help determine the trend of defects found over time. Is the trend improving as the testing phase is winding down, or does the trend remain static or is it worsening? During the AST testing process, we have found defect trend analysis to be one of the more useful metrics to show the health of a project. One approach to showing the trend is to plot the total number of defects over time, as shown in Figure 1.5. [5]

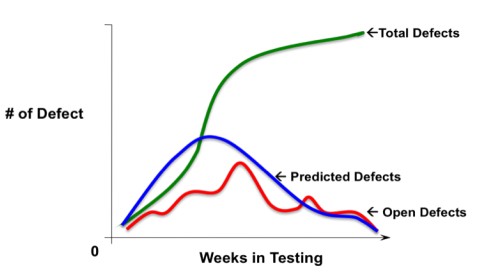


Figure 1.5 Defect trend analysis: total number of defects over time (here after weeks in testing)

Effective defect tracking analysis can present a clear view of the status of testing throughout the project.

**Defect Removal Efficiency**

One of the more popular metrics is *defect removal efficiency* (DRE); this metric is not specific to automation, but it is very useful when used in conjunction with automation efforts. DRE is used to determine the effectiveness of defect removal efforts. It is also an indirect measurement of product quality. The value of the DRE is calculated as a percentage. The higher the percentage, the greater the potential positive impact on the quality of the product. This is because it represents the timely identification and removal of defects at any particular phase.

DT No. of defects found during testing  
 DRE (%) = ------------ = ( -------------------------------------------- )  
 DT + DA No. of defects found during testing +  
 No. of defects found after delivery  
  
DRE = Defect removal efficiency  
DT = Number of defects found during testing  
DA = Number of defects found after delivery

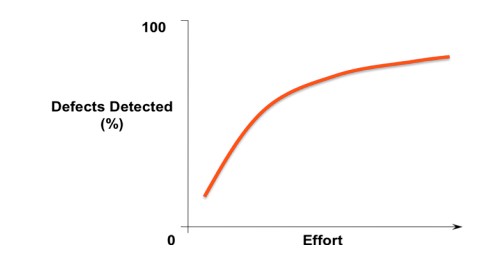


Figure 1.6Defect removal efficiency over the testing lifecycle effort

The highest attainable value of DRE is 1, which equates to 100%. In practice, we have found that an efficiency rating of 100% is not likely. According to Capers Jones, world-class organizations have a DRE greater than 95%. [6] DRE should be measured during the different development phases. For example, low DRE during analysis and design may indicate that more time should be spent improving the way formal technical reviews are conducted.

This calculation can be extended for released products as a measure of the number of defects in the product that were not caught during the product development or testing phases.

**Automated Software Testing ROI**

As we have discussed, metrics help define the progress, health, and quality of an automated testing effort. Without such metrics, it would be practically impossible to quantify, explain with certainty, or demonstrate quality. Along with quality, metrics also help with demonstrating ROI, covered in detail in IAST of this book. ROI measurement, like most metrics, is an ongoing exercise and needs to be closely maintained. Consider the ROI and the various testing metrics when investigating the quality and value of AST. As shown in Figure 1.7, metrics can assist in presenting the ROI for your effort. Be sure to include all facets in your ROI metric as described in IAST.

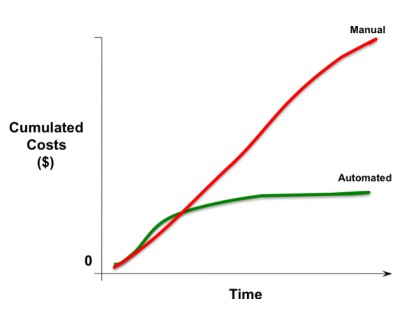


Figure 1.7 AST ROI example (cumulated costs over time)

**Other Software Testing Metrics**

Along with the metrics mentioned in the previous sections, there are a few more common test metrics useful for the overall testing program. Table 1.1 provides a summary and high-level description of some of these additional useful metrics.

**Table 1.1** Additional Common and Useful Software Test Metrics [7]

|  |  |  |
| --- | --- | --- |
| Metric Name | Description | Category |
| Error discovery rate | Number of total defects found/Number of test procedures executed  The *error discovery rate* is used to analyze and support a rational product release decision. | Progress |
| Defect aging | The date a defect was opened versus the date the defect was fixed  The *defect aging* metric indicates turnaround of the defect. | Progress |
| Defect fix retest | The date a defect was fixed and released in a new build versus the date the defect was retested  The *defect fix retest* metric indicates whether the testing team is retesting the fixes fast enough, in order to get an accurate progress metric. | Progress |
| Current quality ratio | Number of test procedures successfully executed (without defects) versus the number of test procedures.  The *current quality ratio* metric indicates the amount of functionality that has successfully been demonstrated. | Quality |
| Problem reports by priority | The number of software problem reports broken down by priority  The *problem reports* metric counts the number of software problems reported, listed by priority. | Quality |

**Root Cause Analysis**

It is not good enough to conduct lessons learned after the AST program has been implemented. Instead, as soon as a problem is uncovered, regardless of the phase or the type of issue - whether it’s a schedule, budget, or software defect problem - a root cause analysis should be conducted, so that corrective actions and adjustments can be made. Root cause analysis should not focus on determining blame; it is a neutral investigation that determines the cause of the problem. For example, a root cause template could be developed where stakeholders can fill in their respective parts if a defect is uncovered in production. The template could list neutral questions, such as, "What is the exact problem and its effect?" "How was it uncovered and who reported it?" "When was it reported?" "Who is/was affected by this problem?" "What is the priority of the problem?" Once all this information has been gathered, stakeholders need to be involved in the discussion and determination of the root cause, how to resolve the issue (corrective action to be taken) and understand the priority of solving and retesting it, and how to prevent that sort of issue from happening in the future.

Defects will be uncovered despite the best-laid plans and implementations; corrective actions, and adjustments are always needed, i.e. expect the unexpected, but have a plan for addressing it Effective AST processes should allow for and support the implementation of necessary corrective actions. They should allow for strategic course correction, schedule adjustments, and deviation from AST phases to adjust to specific project needs, to support continuous process improvement and an ultimate successful delivery.

Root cause analysis is a popular area that has been researched and written about a great deal. What we present here is our approach to implementing it. For more information on root cause analysis and a sample template, review the SixSigma discussion on "Final Solution via Root Cause Analysis (with a Template)." [8]

**Summary**

To assure AST program success, the AST goals need to be not only defined but also constantly tracked. Defect prevention, AST and other software testing metrics, and root causes analysis implementation are important steps to help prevent, detect, and solve process issues and SUT defects. With the help of these steps the health, quality, and progress of an AST effort can be tracked. These activities can also be used to evaluate past performance, current status, and future trends. Good metrics are objective, measurable, meaningful, and simple, and they have easily obtainable data. Traditional software testing metrics used in software quality engineering can be applied and adapted to AST programs. Some metrics specific to automated testing are

* Percent automatable
* Automation progress
* Percent of automated testing coverage
* Software automation ROI (see IAST for more details)
* Automated test effectiveness (related to ROI)

Evaluate the metrics outcome and adjust accordingly.

Track budgets, schedules, and all AST program-related activities to ensure that your plans will be implemented successfully. Take advantage of peer reviews and inspections, activities that have been proven useful in defect prevention.

As covered in IAST, in the test case requirements-gathering phase of your automation effort, evaluate whether it makes sense to automate or not. Given the set of automatable test cases, determine which ones would provide the biggest ROI. Consider that just because a test is automatable doesn’t necessary mean it should be automated. Using this strategy for determining what to automate, you are well on your way to AST success.

**References:**

1. E. Dustin, "The People Problem," www.stpmag.com/issues/stp-2006-04.pdf; "The Schedule Problem," www.stpmag.com/issues/stp-2006-05.pdf; and "The Processes and Budget Problem," www.stpmag.com/issues/stp-2006-06.pdf.
2. www.thefreedictionary.com/metric.
3. Adapted from Dustin et al., *Automated Software Testing*
4. Stephen H. Kan, *Metrics and Models in Software Quality Engineering, 2nd ed.*(Addison-Wesley, 2003).
5. Adapted from www.teknologika.com/blog/SoftwareDevelopmentMetricsDefectTracking.aspx.
6. C. Jones, keynote address, Fifth International Conference on Software Quality, Austin, TX, 1995.
7. Adapted from Dustin et al., *Automated Software Testing*.
8. www.isixsigma.com/library/content/c050516a.asp.

Reference of the full Article: http://www.methodsandtools.com/archive/archive.php?id=94