Acceleration pune

Some real time questions

Q #1) What is Automation?
Answer: Automation is any action that can reduce human efforts.
Q #2) What is Automation testing?
Answer: The process of using special software tools or scripts to perform testing tasks such as entering data, executi ng the test steps and comparing the results, etc. is known as Automation testing.
Q #3) What all things can you automate?
Answer:
Regression test suite
Smoke / Sanity test suite
Build deployment
Test data creation
Automating behind the GUI like testing of APIs and methods.
Q #4) When is Automation testing useful?
Answer: Automation testing is useful in the following scenarios:
a) Regression testing: In case of a bug fix or new module implementation, we have to make sure that the already imp lemented or unchanged functionality is not affected. In this case, we end up running the regression test case multiple times.
For Example: After each change request or bug fix, after each iteration in case of incremental development approach , etc.
b) Non-functional Testing: Testing the non-functional aspects of an application.
For Example, Load testing or performance testing, etc are very difficult for humans to track and analyze.
c) Complex calculation checks or test scenarios that are prone to human errors.
d) Repeated execution of the same tests: Sometimes, we have to run the same set of test case for a different set of dat a or after each build release or on multiple hardware, software or combination of both.
Automating the test cases in the above scenarios helps in achieving the speed of testing and minimizing human error s.
Q #5) How do you identify the test cases which are suitable for automation?
Answer: Identifying the appropriate test cases for automation is the most important step towards automation.
Q #6) Can you achieve 100% automation?
Answer: 100% automation would be difficult to achieve because there would be many edge test cases and some case s that are executed seldom. Automating these cases which are not executed that often will not add value to the autom ated suite.
Q #7) How to decide the tool that one should use for Automation testing in their projects?
Answer: In order to identify the tool for Automation testing in your project:
a) Understand your project requirements thoroughly and identify the testing scenarios that you want to automate.
b) Search for the list of tools that support your project's requirements.
c) Identify your budget for the automation tool. Select the tools within your budget.
d) Identify if you already have skilled resources for the tools. If you don't have the necessary skilled resources then i dentify the cost for training the existing resources or hiring new resources.
e) Now compare each tool for key criteria like:
How easy is it to develop and maintain the scripts for the tool?
Can a non-technical person also execute the test cases with little training?
Does the tool support different types of platforms like web, mobile, desktop, etc based on your project requirements?

Does the tool have a test reporting functionality? If not, is it easily configurable for the tool?
How is the tool for cross-browser support for web-based applications?
How many different testing types can this tool support?
How many languages does the tool support?
f) Once you have compared the tools, select the tool which is within your budget and support your project requireme nts, and gives you more advantages based on the key criteria mentioned above.

Q #8) Currently I do not have any automation in place in my project, but now I want to implement automation, what would be my steps?
Answer:
First, identify which type of testing/test cases you want to automate.
Identify the tool
Design the framework
Create utility files and environment files.
Start scripting
Identify and work on reporting.
Allocating time for enhancing and maintaining the scripts.
Steps required for getting Automation Testing in place for a project include:
Understand the advantages and disadvantages of automation testing and identify the test scenarios which are suitable for automation.
Select the automation tool that is best suited for automating the identified scenarios
Find the tool expert to help in setting up the tool and required environment for executing the test cases using the tool .
Train the team so that they can write scripts in the programming language that the tool supports.
Create the test framework or identify the already existing one that meets your requirements.
Write an execution plan for OS, browsers, mobile devices, etc.
Write programming scripts for manual test cases to convert them into automated test cases.
Report the test case status by using the reporting feature of the tool.
Maintain the scripts for ongoing changes or new features.

Q #9) How do you decide which tool you have to use?
Answer: Concluding which tool is best suitable for the project requires a lot of brainstorming and discussions.

Q #10) Once you identify the tool what would be your next steps?
Answer: Once we finalize the tool, our next step would be to design the framework.

Q #11) What is a framework?
Answer: A framework is a set of the structure of the entire automation suite. It is also a guideline, which if followed can result in a structure that is easy to maintain and enhance.
These guidelines include:
Coding standards
Handling the test data
Maintaining and handling the elements (object repository in QTP)
Handling of environment files and properties file
Reporting of data
Handling logs

Q #12) What are the attributes of a good framework?
Answer: The characteristics include:
Modular: The framework should be adaptable to change. Testers should be able to modify the scripts as per the environment or login information change.
Reusable: The commonly used methods or utilities should be written in a common file that is accessible to all the scripts.
Consistent: The suite should be written in a consistent format by following all the accepted coding practices.
Independent: The scripts should be written in such a way that they are independent of each other. In case one test fails, it should not hold back the remaining test cases (unless it is a login page)
Logger: It is good to have implemented the logging feature in the framework. This would help in case our scripts run for longer hours (say nightly mode), if the script fails at any point of time, having the log file will help us to detect the location along with the type of the error.
Reporting: It is good to have the reporting feature automatically embedded into the framework. Once the scripting is done, we can have the results and reports sent via email.
Integration: Automation Framework should be such that it is easy to integrate with other applications like continuous

integration or triggering the automated script as soon as the build is deployed.

Q #13) Can you do without a framework?
Answer: Frameworks are guidelines and not mandatory rules, so we can do without a framework, but if we create it and follow it, enhancing and maintaining would be easy to implement.

Q #14) What are the different types of the Automation tool that you are aware of?
Answer: Open source tool like Selenium, JMeter, etc.
Paid tools like QTP, Load Runner, Ranorex, RFT, and Rational Robot.

Q #15) What generally is the structure of a framework?
Answer: Normally the structure should have – (It would differ from project to project)
A “src” (source) folder having the actual test scripts.
A”lib” (library) folder having all the libraries and common methods.
A “class” folder having all the class file (in-case using java).
A “log” folder having the log file(s).
A file/folder having all the web element Ids.
A file containing the URL, environment and login information.

Q #16) Where will you maintain information like URL, login, password?
Answer: This information should always be maintained in a separate file.

Q #17) Why do you want to keep this kind of information in a separate file and not directly in the code?
Answer: URL, Login, and passwords are the kind of fields that are used very often and these change as per the envir onment and authorization. In case we hardcode it into our code, we have to change it in every file which has its refer ence.
In case if there are more than 100 files, then it becomes very difficult to change all the 100 files and this, in turn, can lead to errors. So this kind of information is maintained in a separate file so that updating becomes easy.

Q #18) What are the different types of frameworks?
Answer: Different types of frameworks includes:
Keyword-driven framework
Data-Driven framework
Hybrid Framework
Linear Scripting

Q #19) Can you tell some good coding practices while automation?
Answer: Some of the good coding practices include:
Add appropriate comments.
Identify the reusable methods and write it in a separate file.
Follow the language-specific coding conventions.
Maintain the test data in a separate file.
Run your scripts regularly.

Q #20) Any kind of test which you think should not be automated?
Answer:
Tests that are seldom executed.
Exploratory testing
Usability testing
Test which is executed quickly when done manually.

Q #21) Do you think that testing can be done only at the UI level?
Answer: Today as we are moving to the Agile mode, testing is not limited to the UI layer. Early feedback is imperial for an agile project. If we concentrate only on the UI layer, we are actually waiting until the UI is developed and ava ilable to test.

Rather we can test even before the UI is actually developed. We can directly test the APIs or the methods using tools like Cucumber and FitNesse.
In this way, we are giving the feedback much early and are testing even before the UI is developed. Following this approach will help us to test only the GUI aspect of small cosmetic changes or some validations on the UI and will help the developers by giving more time to fix the bugs.

Q #22) How do you select which automation tool is best suited for you?
Answer: Selecting the automation tool depends upon various factors like:
The scope of the application which we want to automate.
Management overhead like cost and budget.
Time to learn and implement the tool.
Type of support available for the tool.
Limitation of the tool

Q #23) What do you think holds the testers back to do automation? Is there a way to overcome it?
Answer: The major hurdle for testers is to learn programming/coding when they want to automate. Since testers do not code, adapting to coding is a bit challenging for testers.
We can overcome it by:
Collaborating with developers when automating.
Considering that automation is the responsibility of the whole team and not only of the testers.
Giving a dedicated time and focus on automation.
Getting proper management support.
You can save these automation testing interview questions as a pdf and print for further reading.

Q #24) What is an Automation testing framework?
Answer: A framework, in general, is a set of guidelines. A set of guidelines, assumptions, concepts and coding practices for creating an execution environment in which the tests will be automated, is known as an Automation testing framework.
An automation testing framework is responsible for creating a test harness with a mechanism to connect with the application under test, take input from a file, execute the test cases and generate the reports for test execution. An automation testing framework should be independent of the application and it should be easy to use, modify or extend.

Q #25) What are the important modules of an automation testing framework?
Answer: Important modules of an Automation testing framework are:
Test Assertion Tool: This tool will provide assert statements for testing the expected values in the application under test. For Example. TestNG, Junit, etc.
Data Setup: Each test case needs to take the user data either from the database or from a file or embedded in the test script. Frameworks data module should take care of the data intake for test scripts and the global variables.
Build Management Tool: Framework needs to be built and deployed for the use of creating test scripts.
Continuous integration tool: With CICD (Continuous Integration and Continuous Development) in place, continuous integration tool is required for integrating and deploying the changes done in the framework at each iteration.
Reporting tool: A reporting tool is required to generate a readable report after the test cases are executed for a better view of the steps, results, and failures.
Logging tool: The logging tool in framework helps in better debugging of the error and bugs.

Q #26) Explain some Automation testing tools.
Answer: Some of the famous Automation testing tools are explained below:
(i) Selenium: Selenium is a test framework for web application automation testing. It supports multiple browsers and is OS independent. Selenium also supports various programming languages like Java, C#, PHP, Ruby, and Perl, etc.

Selenium is an open-source set of libraries which can be used to develop additional test frameworks or test scripts for testing web-based applications.
(ii) UFT: Unified Functional Testing is a licensed tool for functional testing. It provides a wide range of features like APIs, web services, etc and also supports multiple platforms like desktops, web, and mobile. UFT scripts are written in visual basic scripting language.

(iii) Appium: Appium is an open-source mobile application testing tool. It is used to automate testing on cross-platform, native, hybrid and web-based mobile applications. Appium automates any mobile application from any language with full access to APIs and DBs from the test code.
Appium is based on the client-server architecture and has evolved from selenium.
(iv) Cucumber: Cucumber is an open-source behavior-driven development tool. It is used for web-based application automation testing and supports languages like ruby, java, scala, groovy, etc. Cucumber reads executable specification written in plain text and tests the application under test for those specifications.
For cucumber to understand the scenarios in plain text, we have to follow some basic syntax rules which are known as Gherkin.
(v) TestComplete: TestComplete is a licensed automated UI testing tool to test the application across different platforms like desktops, web, mobile, etc. It provides flexibility to record a test case on one browser and run it on multiple browsers and thus supports cross browsers testing.
TestComplete has inbuilt object recognition algorithm which uniquely identifies an object and stores it in the repository.

Q #27) What are the different types of testing framework techniques?
Answer: There are four types of automation testing framework techniques.
They are:
(i) Modular Testing framework:
This framework is built on the concept of abstraction. In this framework, the tester creates scripts for each module of the application under test individually and then these scripts are combined in the hierarchical order to create large test cases.
It creates an abstraction layer between the modules, thus any modifications in test scripts for one module do not affect any other modules.
Advantages of this framework:
Easier maintenance and scalability of test cases.
Creating test cases by using already scripted modules is easier and faster.
Disadvantages:
Test cases have data embedded in them. Thus executing the same test script with different data is a big change at the script level.
(ii) Data-Driven Testing framework:
In the Data-driven testing framework, the input data and the expected output data corresponding to the input data is stored in a file or database and the automated script runs the same set of test steps for multiple sets of data. With this framework, we can run multiple test cases where only the input data differs and the steps of execution are the same.
Advantages:
Reduces the number of test scripts that are required to be executed. We execute the same script multiple times with different data.
Less coding for automation testing.
Greater flexibility for maintaining and fixing the bugs or enhancing the functionality.
Test data can be created even before the automated system for testing is ready.
Disadvantages:
Only similar test cases with the same set of execution steps can be combined for multiple sets of data. The different set of execution steps require a different test case.
(iii) Keyword-Driven Testing framework:
It is an application-independent testing framework that uses data tables and self-explanatory keywords. Keywords explain the actions to be performed on the application under test and the data table provides the input and expected output data.
Keyword-based testing is an increment to data-driven testing.
Advantages:
Less coding and the same script can be used for multiple sets of data.
Automation expertise is not required for creating a test case using the already existing keywords for actions.
Same keywords can be used across multiple test cases.
Disadvantages:
This framework is more complicated as it needs to take care of the keyword actions and also the data input.
Test cases get longer and complex thereby affecting the maintainability of the same.

(iv) Hybrid Testing framework:
This framework is a combination of all the above-mentioned testing frameworks (Modular, Data-driven, and Keywo rd-driven).
In this framework, the test cases are developed from modular scripts by combining them in the modular testing fram ework. Each of the test cases uses a driver script that uses a data file as in the data-driven framework and a keyword-based action file.
Advantages:
Modular and easy to maintain.
Less coding can take care of more test cases.
One test case can be executed with multiple sets of data.
Disadvantages:
Complex to read, maintain and enhance.

Q #28) When do you prefer Manual testing over Automation testing?
Answer: We prefer manual testing over automation testing in the following cases:
The project is short-term and writing scripts will be time-consuming and costly when compared to manual testing.
Flexibility is required. Automated test cases are programmed and run in a specific way of configurations.
Usability testing needs to be performed.
Applications/module is newly developed and has no previous test cases.
Ad-hoc or exploratory testing needs to be performed.

Q #29) Is Automation testing in agile Methodology useful or not?
Answer: Automation testing is useful for regression, smoke or sanity testing. All these types of testing in the traditio nal waterfall model happen at the end of the cycle and sometimes if there are not many enhancements to the applicati on, we might not even have to do regression testing.
Whereas, in agile methodology, every iteration requires executing the regression test case as some new functionalitie s is added.
Also, the regression suite itself keeps growing after each sprint as the functional test cases of the current sprint modu le need to be added to the regression suite for the next sprint.
Thus, Automation testing in agile methodology is very useful and helps in achieving maximum test coverage in less time of the sprint.

Q #30) List some advantages and disadvantages of Automation testing.
Answer:
Advantages:
Fewer human resources
Reusability
More Test Coverage in less time
Reliability
Parallel execution of test cases
Fast
Disadvantages:
Development and maintenance time is more.
Tool Cost
Skilled resources are required.
Environment setup
Test Script debugging is an issue.

Q #31) List some advantages and disadvantages of Manual testing.
Answer:
Advantages:
No environment setup needed.
Programming knowledge is not required.
Recommended for dynamically changing requirements.
Allow human observation power to detect more bugs.

The cost is less for short-term projects.
Flexibility
Disadvantages:
Difficult to perform complex calculations.
Reusability
Time taking
High risk of human errors or mistakes.
More human resources are required.

Q #32) Can we do Automation testing without a framework? If yes, then why do we need a framework?
Answer: Yes, We can perform automation testing even without using a framework. We can just understand the tool t hat we are using for automation and program the steps in the programming language that tools support.
If we automate test cases without a framework then there won't be any consistency in the programming scripts for te st cases.
A framework is required to give a set of guidelines that everyone has to follow to have maintained readability, reusa bility, and consistency in the test scripts. A framework also provides one common ground for reporting and logging f unctionality.

Q #33) How will you automate basic "login" functionality test cases for an application?
Answer: Assuming that the automation tool and framework is already in place of the test environment.
To test the basic "Login" functionality:
Understand the project requirement: Login functionality will have a username textbox, a password textbox, and a log in button.
Identify the Test scenarios: For login functionality, the possible test scenarios are:
Blank username and password
Invalid username and password
A valid username and invalid password
Valid username and password
Prepare a Data input file with the data corresponding to each scenario.
Launch the tool from the program.
Identify the username field, password field, and the login button.
For each test scenario, get the data from the data file and enter into the corresponding fields. Program click on the lo gin button after entering the data.
Validate the error message for negative scenarios and the success message for positive scenarios in the test script wit h the help of assertions.
Run the test suite and generate the report.

Q #34) Is Automation testing a Black box testing or White-box testing?
Answer: Automation testing is mostly a black box testing as we just program the steps that a manual tester performs for application under test without knowing the low-level design or code of the application.
Sometimes, automated test scripts need access to the database details that are used in the application under test or so me more coding details and thus can be a type of white-box testing.
Thus automated testing can be both black or white box type of testing depending on the scenarios in which automati on is performed.

Q #35) How many test cases have you automated per day?
Answer: Well, the number depends on the complexity of the test cases. When the complexity was limited, I was able to automate 5 to 6 test cases per day. Sometimes, I was able to automate only one test case for complex scenarios.
I have also broken down my test cases into different components like, take input, do the calculation, verify the outpu t etc. in case of very complex scenarios and have taken 2 or more days.

Q #36) What factors determine the effectiveness of Automation testing?
Answer: Some of the factors that determine the effectiveness of automation testing are:
Time saved by running scripts over the manual execution of test cases.
Defects found

Test Coverage or code coverage
Maintenance time or development time
Stability of the scripts
Test Reusability
Quality of the software under test

Q #37) Which test cases can be automated?
Answer: Types of test cases which can be automated are:
(i) Smoke test cases: Smoke testing is also known as build verification testing. Smoke test cases are run every time when a new build is released to check the health of the build for acceptance to perform testing.
(ii) Regression Test Cases: Regression testing is the testing to ensure that previously developed modules are function ing as expected after a new module is added or a bug is fixed.
Regression test cases are very crucial in incremental software approach where a new functionality is added at each in crement phase. In this case, regression testing is performed at each incremental phase.
(iii) Complex Calculation test cases: Test cases which involve some complex calculations to verify a field for an app lication fall into this category. Complex calculation results are more prone to human errors hence when automated th ey give accurate results.
(iv) Data-driven test cases: Test cases which have the same set of steps and run multiple times with the change of dat a are known as data-driven test cases. Automated testing for these kinds of test cases is quick and cost-effective.
(v) Non-functional test cases: Test cases like load tests and performance tests require a simulated environment with multiple users and multiple hardware or software combinations.
Setting up multiple environments manually is impossible for each combination or number of users. Automated tools can easily create this environment to perform non-functional testing easily.

Q #38) What are the phases in Automation testing Life Cycle?
Answer: The phases in Automation testing life Cycle include:
The decision to perform automation testing.
Identify and learn about the automation tool.
Determine the scope of automation testing.
Design and develop a test suite.
Test Execution
Maintenance of test scripts.

Q #39) What is an Automated test script?
Answer: An automated test script is a short program that is written in a programming language to perform a set of in structions on an application under test to verify if the application is as per the requirements.
This program when run, gives the test results as pass or fail to depend on if the application is as per the expectations.

Conclusion
These are the main questions that are independent of the automation tool or programming language. Automation test ing interviews also includes tool and programming language-specific questions depending upon the tool that you hav e worked with.
Most of the test automation interview questions are centered on the framework you develop, so it is recommended th at you create and understand your test framework thoroughly. When I am interviewing, and the candidate has answer ed my question on the framework, I also prefer asking a language-specific question (core java in my case).
The questions start from basics of java to write the logic of some basic scenario like:
How would you extract a set of text from a given line?
How would you extract the URL?
In any web page, at any frame, the number of links and its content change dynamically, how would you handle it?
How do you handle images and flash objects?
How do you find a word in a line?
Answers to all these test automation interview questions are very much specific to the tool/language that you are usi ng for automating. So before you go for the interview, brush up your programming skills.
In case you did not get a chance to create your framework and someone else has created it, then make some time to u nderstand it thoroughly before sitting for the interview.

Some tips for automation testing interviews would be:
Know your tool thoroughly.
Learn the locator techniques used by your tool.
Practice programming using the language which you use for automation testing.
Learn your framework and its components.
It's always advantageous if you have been involved in the development of your framework. So, be thorough with the modules in the framework which you have worked on.