**Question 1. What Is Mobile Application Testing And How Is It Different From Mobile Testing?**

[Mobile Application Testing](https://www.edureka.co/blog/mobile-application-testing/) (MAT) is the testing of an application on mobile devices and it is different from Mobile Testing (MT) in the term that, in MT we focus on the native application features of Mobile devices like Call, SMS, Media Player, etc. Meanwhile, in MAT we focus only on the functionality & features of the application under Test.

**Question 2. Explain The Difference Between Simulator And Emulator?**

Emulation is the process of mimicking the outwardly observable behavior to match an existing target. The internal state of the emulation mechanism does not have to accurately reflect the internal state of the target which it is emulating.

Simulation, on the other hand, involves modeling the underlying state of the target. The end result of a good simulation is that the simulation model will emulate the target which it is simulating.

Ideally, you should be able to look into the simulation and observe properties that you would also see if you looked into the original target. In practice, there are some shortcuts to the simulation for performance reasons — that is, some internal aspects of the simulation may actually be an emulation.

**Question 3. List out the types of mobile app testing.**

The types of mobile app testing include:

* Usability testing
* Compatibility testing
* Interface testing
* Services testing
* Low-level resource testing
* [Performance testing](https://www.edureka.co/blog/performance-testing-tutorial/)
* Operational testing
* Installation tests
* Security testing

**Question 4. Explain the general structure of mobile application testing frameworks?**

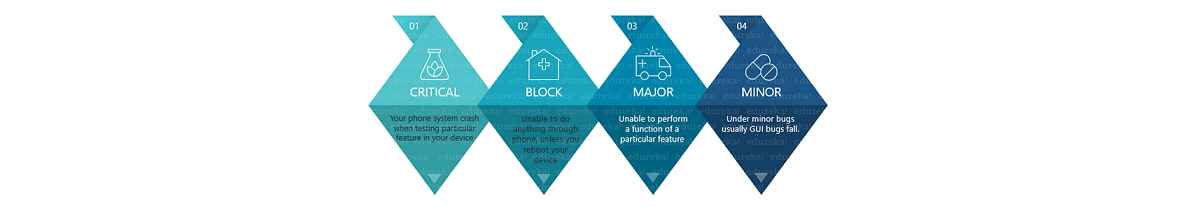
Mobile application testing framework includes three segments:

**Application Package:** It is the target application that requires to be tested.

**Instrumentation TestRunner:** It is a test case runner that runs test cases on the target application. It includes an SDK tool for building test and a tool that provides APIs for writing a program that controls an Android device, for example, MonkeyRunner.

**Test Package:** It includes two classes, Test case classes, and Mock objects. Test case classes include test methods to perform on the target application, while the mock object includes mock data that will be used as sample input for test cases.

**Question 5. Mention what are the common bugs found while mobile testing?**

**Question 6. Full form of various application extensions.**

* **iPA:**iOS APP Store Package
* **APK:**Android Application Package file
* **exe:**Executable File
* **jad:**[Java Application](https://www.edureka.co/blog/applications-of-java/) Descriptor
* **prc:**Palm Resource Compiler

**Question 7. Mention the Different Types of Mobile Applications.**

Mobile applications can be broadly categorized into three categories i.e, Native app, Web app and Hybrid App.

**Native App:** Native app is developed specifically for one platform, which is coded with a specific programming language (like Objective C for iOS, [Java](https://www.edureka.co/blog/java-tutorial/) for Android) and installed directly onto the device and can take full advantage of all the device features. Native apps can use the device’s notification system and can work offline. Native apps are installed through an application store (such as Google Play or Apple’s App Store). Native mobile apps provide fast performance and a high degree of reliability. Example of native apps: Temple Run, Candy Crush, etc.

**Web App:**Web applications are mobile web portals that are designed, customized and hosted specifically for mobiles. They are accessed through the mobile device’s web browser using a URL. Web apps became really popular when HTML5 came around and people realized that they can obtain native-like functionality in the browser. Mobile web applications cannot use device functionality. Example of web app: google.com, m.snapdeal.com, m.yahoo.com, etc.

**Hybrid App:**Hybrid Apps are web apps embedded in a native app, run on the device, and are written with web technologies (HTML5, CSS, and [JavaScript](https://www.edureka.co/blog/what-is-javascript/)). Hybrid apps run inside a native container and leverage the device’s browser engine (but not the browser) to render the HTML and process the JavaScript locally. A web-to-native abstraction layer enables access to devices capabilities that are not accessible in mobile web applications, such as the accelerometer, camera, and local storage. A hybrid app is **NOT** tied to any platform or any particular mobile device. So, it can run on any device once built. Example of a Hybrid app: Flipkart, Facebook, Twitter, etc.

**Question 8. What is the Appium philosophy?**

These are the four philosophies [Appium](https://www.edureka.co/blog/what-is-appium/" \t "_blank) is based around-

* Test the same app you submit to the marketplace
* Write your tests in any language, using any framework
* Use a standard automation specification and API
* Build a large and thriving open-source community effort

**Question 9. What Is Appium’s Strongest Point in your opinion?**

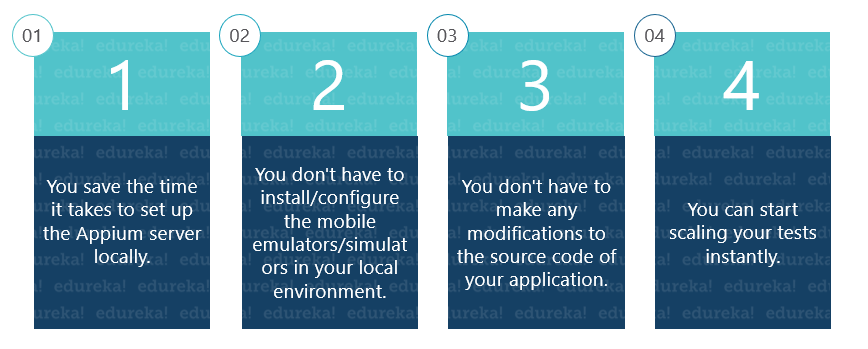
Appium is based on [Selenium](https://www.edureka.co/blog/selenium-tutorial) which is an HTTP protocol by Google designed to automate browsers. The idea is actually very nice as automating an app (especially a webview-based one) is not so different (in terms of required APIs) from automating a browser.

[Appium is also designed to encourage a 2-tier architecture](https://www.edureka.co/blog/appium-architecture/): a machine runs the test written in one language (C#, [Ruby](https://www.edureka.co/blog/ruby-on-rails-tutorial/), JavaScript are only a few among the many supported ones) and another one (the test server) actually executes it. Furthermore the WebDriver protocol targets scalability (because based on HTTP), this makes Appium very scalable as well; remember that you will need to write your test once, Appium will be in charge of executing it on more platforms.

**Question 10. What Is Appium?**

Appium is an open source, cross-platform [automation](https://www.edureka.co/blog/rpa-tutorial/) testing tool. It is used for automating test cases for native, hybrid and web applications. The tool has a major focus on both Android and [iOS apps](https://www.edureka.co/blog/swift-tutorial) and was only restricted to the mobile application testing domain. Recently, a few updates back, Appium also announced that they will support the testing of desktop applications for windows. Appium is developed and maintained by Sauce Labs. Currently, Appium version 1.10 is being distributed. Appium first started off as a command line based testing service that can be installed using [Node.js](https://www.edureka.co/blog/nodejs-tutorial/). In their latest release, named ‘Appium desktop’ they have released a robust and refined tool with an intuitive graphical user interface.

**Question 11. What are the main advantages Of Using Appium On Sauce Labs?**



**Question 12. List Out The Appium Abilities.**

**Appium abilities are:**

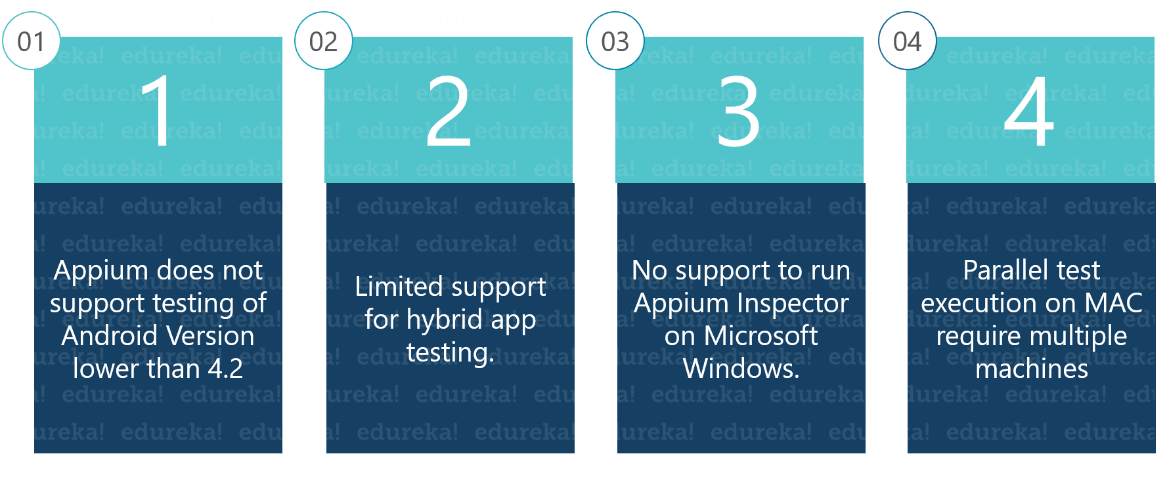
* Test Web
* Provides cross-platform for Native and Hybrid mobile automation
* Support JSON wire protocol
* It does not require recompilation of App
* Support automation test on a physical device as well as simulator or emulator both
* It has no dependency on a mobile device

**Question 13. Can you comment about the performance of Appium?**

Appium is not a huge application and requires very little memory. Its architecture is actually pretty simple and light as Appium acts like a proxy between your test machine and each platform automation toolkit. Once up and running, Appium will listen to HTTP requests from your tests. When a new session is created, a component in Appium’s Node.js code called *\_proxy\_* will forward these Selenium commands to active platform drivers.

In the case of Android, for example, Appium will forward incoming commands to the [chromedriver] (90% of cases, Appium will not even change commands while routing them), this happens because [ChromeDriver supports WebDriver and Selenium](https://www.edureka.co/blog/selenium-chromedriver-and-geckodriver/). For this reason, Appium will not allocate much memory itself, you will see a lot of memory being allocated by other processes like [adb], ChromeDriver or the iOS automation toolkit (called by Appium while testing and automating).

**Question 14. List Out The Limitations Of Using Appium?**



**Question 15. Do you Need A Server Machine To Run Tests On Appium?**

No! Appium promotes a 2-tier architecture where a test machine connects to a test server running Appium and automating the whole thing. However this configuration is not mandatory, you can have Appium running on the same machine where your test runs. Instead of connecting to a remote host, your test will connect to Appium using the loopback address.

**Intermediate Questions**

**Question 16. What Type Of Tests Are Suitable For Appium?**

When it comes to testing, especially webview-based apps, there are a lot of scenarios that can be tested also depending on the feature coverage you want to ensure. Appium is pretty handy for testing scenarios that users will go through when using your app.

But, if you need to test more than UX simple interactions, then Appium will become a limitation. Think about features like keyboarding. It is not so easy when complex touch/keyboard mixed scenarios are involved, the probability of a false failure is high; do not misunderstand me on this: I am not saying it is impossible to do, just not so easy as you might think!

Another little nightmare with Appium is exchanging data. When your test needs to exchange data with your app (especially in the incoming direction), you will need to play some tricks. So always consider that sending and receiving information is not that straightforward. It is not Appium’s fault, the WebDriver specification was designed for automating stuff, not exchanging data!

**Question 17. List some issues Faced With Cross-Platform Testing?**

Generally, the issue depends upon the different OS/device version. It might be that the same application is working on one OS while it might not work on another version. For example, we faced an issue that our application was working fine on iOS 6.x version devices but on tapping a few modules on iOS 5.x devices application crashes and the same happened with 2.3.5 Vs.

**Question 18. Explain The Design Concept Of Appium.**

* Appium is an “HTTP Server” written using Node.js platform and drives iOS and Android session using Webdriver JSON wire protocol. Hence, before initializing the Appium Server, Node.js must be pre-installed on the system
* When Appium is downloaded and installed, then a server is set up on a machine that exposes a REST API.
* It receives connection and command request from the client and executes that command on mobile devices (Android / iOS)
* It responds back with HTTP responses. Again, to execute this request, it uses the mobile [test automation frameworks](https://www.edureka.co/blog/test-automation-frameworks/) to drive the user interface of the apps.
* Apple Instruments for iOS (Instruments are available only in Xcode 3.0 or later with OS X v10.5 and later).
* Google UIAutomator for Android API level 16 or higher
* Selendroid for Android API level 15 or less

**Question 19. I Already Have Platform-specific Tests For My App, What Should I Do To Migrate To Appium?**

Unfortunately, there is no magic formula to translate your tests into Selenium tests. If you developed a test framework on different layers and observed good programming principles, you should be able to act on some components in your tests in order to migrate your suites to Appium.

Your current tests are going to be easy to migrate if they are already using an automation framework or something close to a command-based interaction. Truth be told, you will probably need to write your tests from the beginning.

**Question 20. When Performing End To End Mobile Testing What Major Criteria Are Taken Into Consideration?**

Major areas are

* Installation of the application.
* First time launching an application without having network access.
* Uninstallation of the app.
* The orientation of the app if it is supported.
* Testing application performance on different kinds of devices and network scenarios.
* Testing the application response and how it responds when an invalid user credential is provided.

Also, if your application is accessing any network then you must check the logs generated during that period so that the sensitive information should always go in an encrypted form.

**Question 21. How Do You Test Patches Intended For An App Already In Production?**

We generally do [regression testing](https://www.edureka.co/blog/regression-testing) of a relative module and mainly focus on the area which is related to the bug fixes as per the developer. This is because we cannot perform complete regression testing in a very short span of time. So, we just perform [sanity tests](https://www.edureka.co/blog/smoke-testing-and-sanity-testing) on the rest of the application, but only for high priority devices. High priority devices are systems that are running the latest versions of the operating system.

**Question 22. What Kind of Testing Would You Perform For a General Application?**

The very first test we have to perform is installation. After that, we check the basic functionality and following which, we check the connectivity of the application with its peripherals. Then we uninstall the build and verify how the application responds when we interrupt it during installation. We also check interruption scenarios when our application requests a network call.

We also perform low network/poor connectivity testing during a network call. The process of upgrading from an older version to a newer version is also tested. Navigation in the application without a network is an important feature that is tested in general testing. Also, the compatibility of the app on different kinds of phones is a major criterion in general testing.

**Question 23. How Much Time Does It Take To Write A Test In Appium?**

Of course, it depends on the test. If your test simply runs a scenario, it will take as many commands as the number of interactions needed to be performed (thus very few lines). If you are trying to exchange data, then your test will take more time for sure and the test will also become difficult to read.

**Question 24. What Test Frameworks Are Supported By Appium?**

Appium does not support test frameworks because there is no need to support them. You can use Appium with all sorts of testing frameworks. NUnit and .NET Unit Test Framework are just a few examples. You will write your tests using one of the drivers for Appium, thus your tests will interface with Appium just in terms of an external dependency.

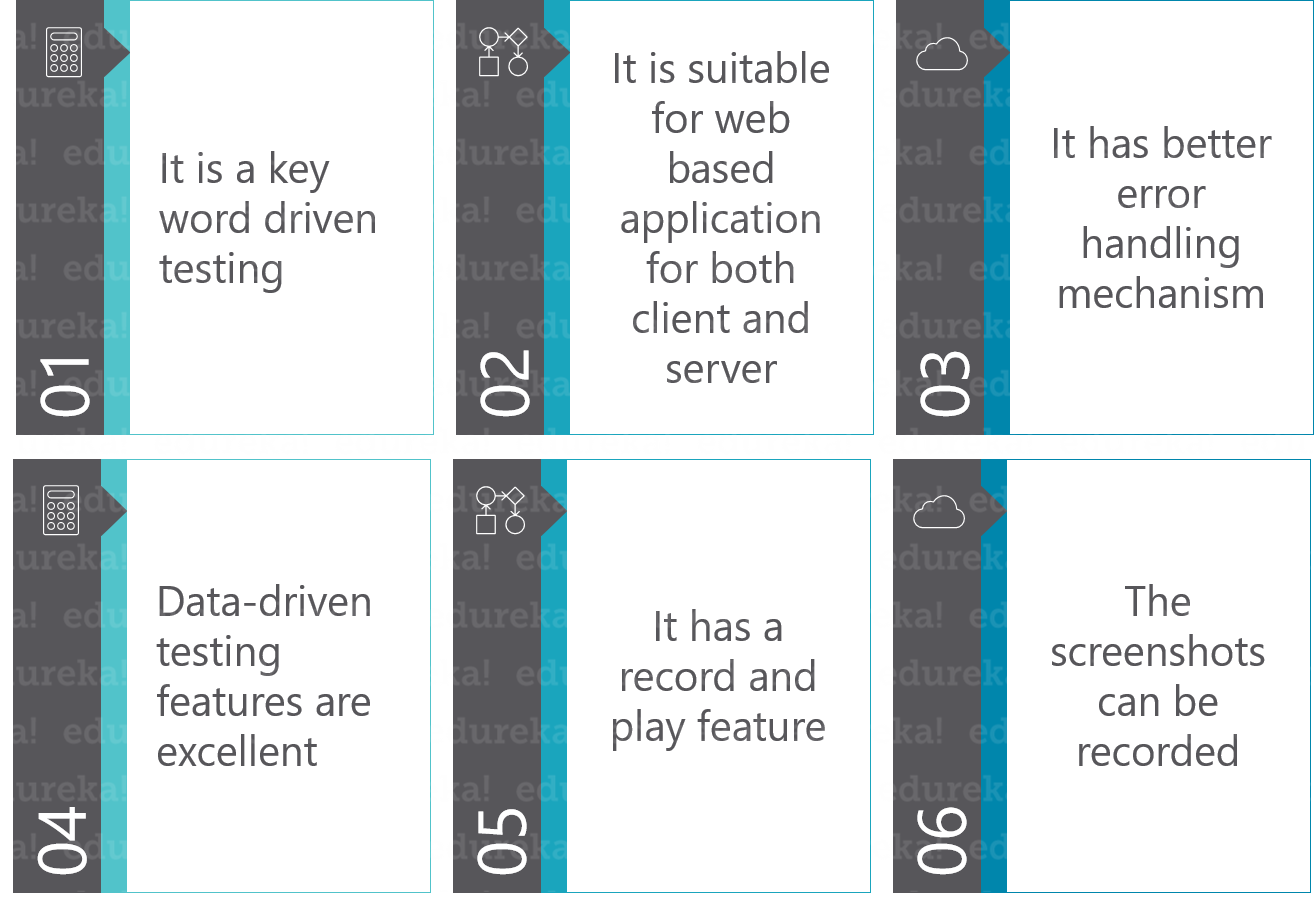
**Question 25. What Is Data Exchange?**

When I say “data exchange” I am not referring to scenarios like getting or setting the value of a textbox or an elements attribute. All these things are easy to achieve in Appium as Selenium provides commands just for those. By “data exchange” I mean exchanging information hosted by complex objects stored in different parts of your webview-based app like the window object.

Consider the scenario when you dispatch and capture events. Your app can possibly perform many functions and the way the flow of data can be handled are many. Some objects might also have a state and the state machine behind some scenarios in your app can be large and articulated. For all these reasons you might experience problems when testing.

**Question 26. What Are The Features And Benefits Of Quick Test Pro(qtp)?**

**The following are the features and benefits of Quick Test Pro:**



**Question 27. Consider A Scenario Where You Don’t Want To Setup A Whole Infrastructure, And Neither Spend Money. Can Appium Help Out In Such A Scenario?**

If you think about it, what really is required from is the test script. The fact that you must deploy an Appium server somewhere is just an extra feature. If you want to skip this part, you can rely on some web services that have already deployed a whole architecture of Appium servers for your tests. Most of them are online labs and they support Selenium and Appium. In this way, you don’t spend exorbitant amounts of money and you also don’t invest time and effort on new infrastructure.

**Question 28. Is Debugging Appium Difficult?**

No. Appium is a Node.js application, therefore, in essence, it is Javascript. The code is available on GitHub and can be downloaded in a few seconds as it is small and not so complex. Depending on what you have to debug, you will probably need to go deeper in your debugging experience. However, there are some key points were setting a breakpoint is always worth it, for example, the proxy component is worth a mention. In **appium/lib/server/proxy.js** you can set a breakpoint in function **doProxy(req, res)**, that will be hit every time commands are sent to platform-specific components to be translated into automation commands.

**Question 29. Mention The Basic Requirement For Writing Appium Tests.**

**For writing Appium tests you 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 them to the Appium server over HTTP.
* **Appium Session:** You have to first initialize a session, as such Appium test takes place in that session. Once the Automation is done for one session, it can be ended.
* **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.

**Question 30. What Are The Risks Associated In Automation Testing?**

**The risks of Automation Testing are:**

* **Do you have skilled resources?** Automation testing demands resources with some knowledge about programming. Firstly, focus on resources and then identify whether the resources have proper knowledge for automation testing. Are they capable to adapt easily to the new technologies? These measures are to be well assessed for building an automation testing team.
* **The initial cost for automation is high.**The initial cost for automation is too high for the initial setup. It includes the cost of automated tools to be purchased, training and maintenance of the test scripts. The unsatisfied customer base is high for automation testing their products. It should be ensured that the cost compensates the testing results.
* **If UI is not fixed, do not think about automation:**Prior to automating the user interface, it should be strongly be determined that, whether the UI is changing extensively or the cost of the automated script maintenance is high or not.
* **Stop automating the tests which run once**: Ensure that certain test cases might be running once and not included in the regression testing. Avoid automating such test modules.

**Question 31. How Can I Run Android Tests Without Appium?**

For older versions of Android, Appium might not be supported. For instance, Appium is only supported for Android versions 4.4 or later for Mobile Web Application tests, Android versions 2.3, 4.0 and later for Mobile Native Application and Mobile Hybrid Application tests.

For those versions in which Appium is not supported, you can request an emulator driven by Webdriver and Selendroid. All you need to do is, use the Platforms Configurator and select Selenium for the API instead of Appium. In the Sauce Labs test, you will notice that the top of the emulator says “AndroidDriver Webview App”. In addition, you will notice that you will get a “Selenium Log” tab which has the output of the Selendroid driver.

With an emulator driven by Webdriver & Selendroid, you will be able to test Mobile Web Application only. You should be able to select any Android emulator version from 4.0 to the latest version and any Android emulator skin (e.g “deviceName”: “Samsung Galaxy Tab 3 Emulator”).

**Question 32. How Can I Run iOS Tests Without Appium?**

For older versions of iOS, Appium might not be supported. For instance, Appium supports foriOS versions 6.1 and later. For earlier versions of iOS, the tool or driver used to drive your mobile applications automated test is called iWebdriver.

To obtain a simulator driven by iWebdriver use the Platforms Configurator and select Selenium for the API instead of Appium. With an emulator driven by iWebdriver, you will be able to test Mobile Web Application only. In addition, in the Sauce Labs test, you will notice a “Selenium Log” tab which has the output of iWebdriver.

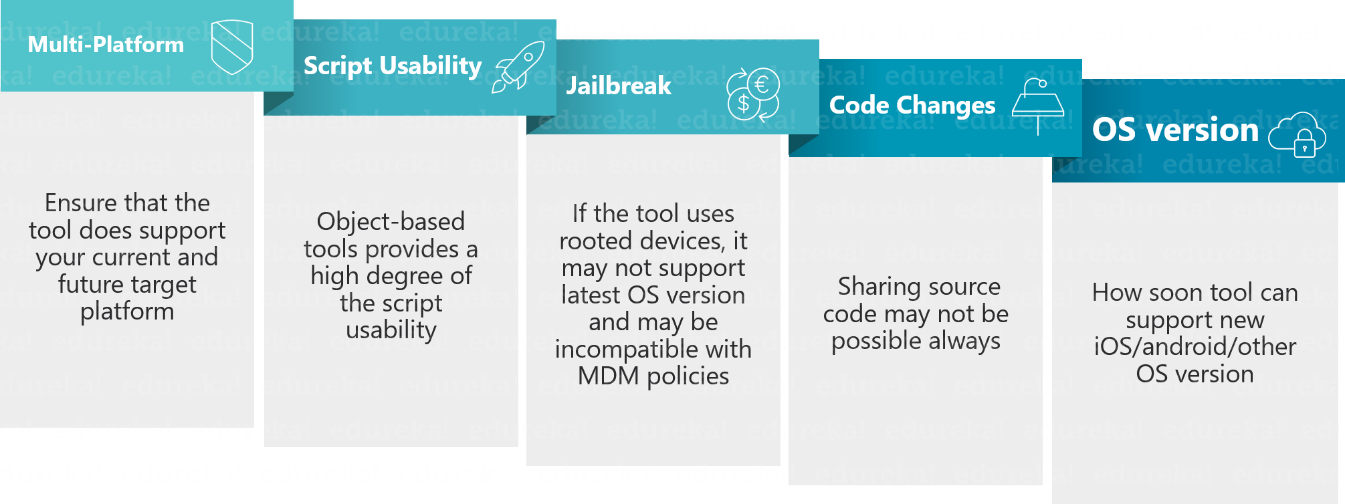
**Question 33. Can You Explain The Filters That You  Create While Checking Logs?**

Filters help you in finding relevant information about your application and you can create a filter based on the application package name like *com.abc.com.*You can save this filter by any name. When you click on this filter, you will see only those logs which are from your application.

You can create a filter based on**log tag** which is related to a  particular line of code, for example, if you have placed *system.out* to print the output then you can create a filter by a tag *System.out.* Then it will shortlist all the print output in the code. You can create a filter by **Choreographer** which helps in finding the skipped frames if you want to see them. You can create filters corresponding to your **PID** and **log message**.

**Question 34. Mention The Selecting Criteria For Test Automation Tool For Mobile Testing.**

For mobile testing, the[test automation tool](https://www.edureka.co/blog/software-testing-tools/#AutomationTestingTools) should have the following criteria

****

**Question 35. List Out The Most Common Problem That A Tester Faces While Performing Mobile Testing In Cloud Computing?**

**Challenges that a tester faces while performing mobile testing are:**

* Subscription model
* High Costing
* Lock-in
* Internet connectivity issues
* Immense consumption of time in image-based automation testing.
* Automation is restricted by the framework

**Question 36: What Would You Prefer To Test Your Application On – Real Devices Or Use Simulators/Emulators?**

It’s one of the most commonly asked Appium interview questions. You’ve to be a little logical and practical while answering it. Don’t just simply answer “It would depend on what you need.”, because it would be a layman answer which interviewer won’t expect from you. Rather you should explain it by an example. You can answer by saying that it’s always best to test on real devices. As it would allow you to catch errors that you may not detect otherwise, but, you have to configure the device smartly with the Appium server so that it can detect the device. Sometimes the ADB(Android Debug Bridge) may disconnect from the device even if it remains plugged in, and it can cause your tests to fail. To handle such issues, you can write a module which resets the ADB after some time to re-connect to the devices.

**Question 37: What Is Appium Inspector And Why Is It Used?**

It is similar to the [Selenium IDE](https://www.edureka.co/blog/selenium-ide) plugin and enables the record and playback support in Appium. It captures the moves of a native application by inspecting DOM and produces the test scripts in any desired language. However, Appium Inspector does not support Windows and uses the **<*UIAutomator*>** viewer.

**Question 38: What Are The Probable Errors You Might Come Across While Working With Appium?**

The following are the errors you might observe with Appium.

* **Error#1:**Missing desired capabilities e.g. Device Name, PlatformName.
* **Error#2:** Couldn’t locate **ADB**. You may have missed setting the **<*ANDROID\_HOME*>** environment variable.
* **Error#3:** Selenium exception **<*openqa.selenium.SessionNotCreatedException*>**. It indicates a failure in creating a new session.
* **Error#4:** Failure in locating a DOM element or determining the [XPath](https://www.edureka.co/blog/xpath-in-selenium/).

**Question 39. What Are The Problems Faced While Running A Test In A Multithreaded Environment?**

You need special care while using Appium in a multithreaded environment. The problem does not rely on the fact of using threads in your tests. You can use them but, you must ensure that no more than one test runs at the same time against the same Appium server. Appium does not support multiple sessions, and unless you have implemented an additional layer on top of it to handle this case, multiple tests might fail.

**Advanced Questions**

**Question 40: List The Selenium Commands That Work With Appium.**

There are a number of Selenium commands that work with Appium tool.

* Locate commands using ID or class names.
* Raise events on elements **e.g. Click()**.
* Text commands like **type()**.
* **Get/Set** element properties.
* Commands to run JavaScript.
* Switch context between different web views like switching **<*iFrames*>** in Selenium Webdriver.
* Commands to manage alert boxes

Related Article: [Automation Interview Questions](https://www.edureka.co/blog/interview-questions/test-automation-interview-questions)

**Question 41: List OneThing Which You Cannot Do With Emulators But You Can Do With A Real Device.**

You can test the interrupts like

* Phone calls & Messages
* Battery drains out while using the application under test
* Low battery scenarios
* Memory card mount/unmount scenarios
* Actual performance of your application
* Bluetooth related testing.

**Question 42: Which Tools Are For Performance Testing And Automation?**

For performance testing of web services, you can use [JMeter](https://www.edureka.co/blog/jmeter-tutorial/). It is an open source tool which can be used to test the API’s performances.

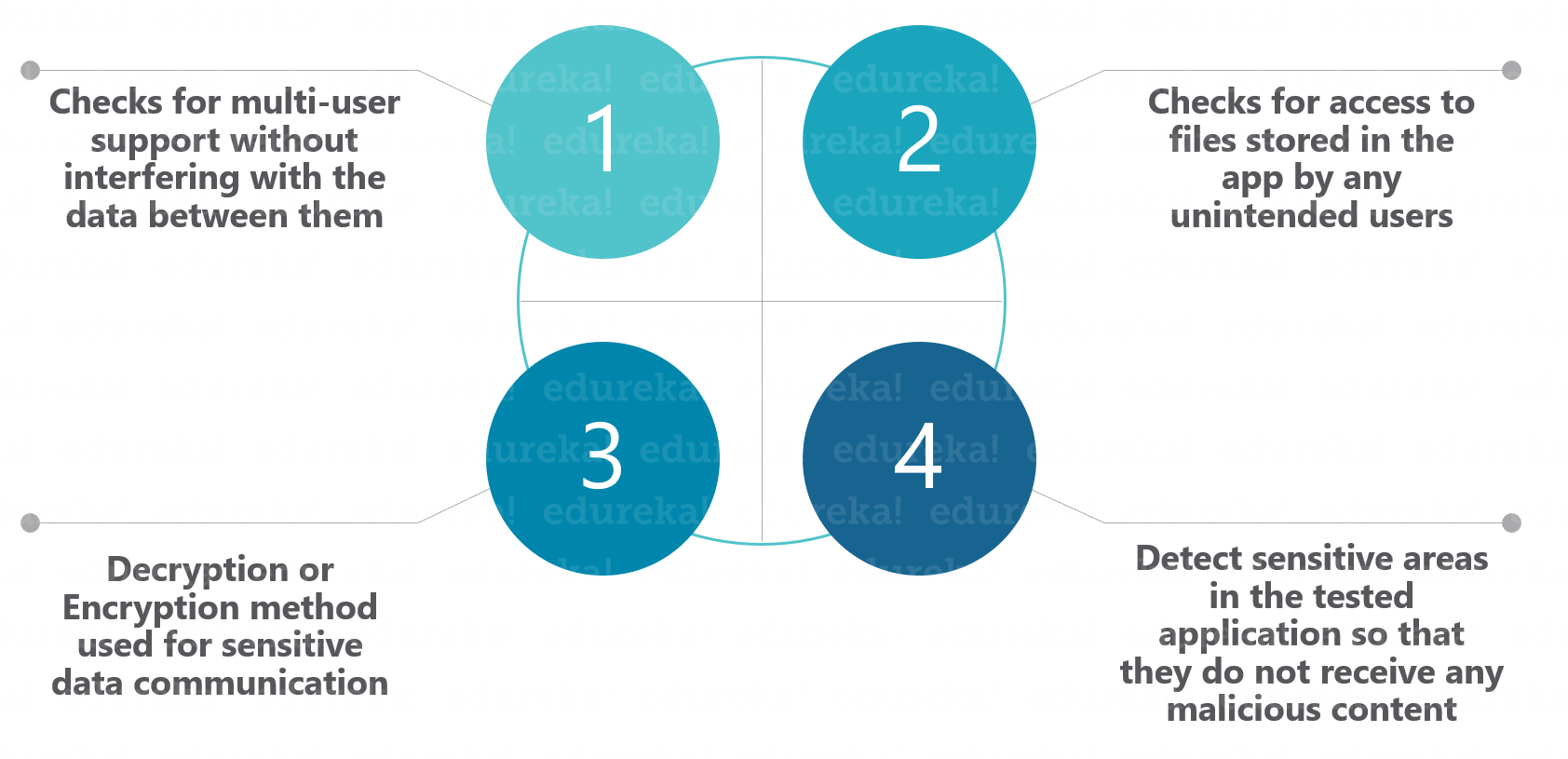
For automation, there are several paid tools available in the market like SeeTest, Ranorex, Silk Mobile etc while free automation tools are Calabash, Appium, Robotium for Android, KIF for iOS.

**Question 43: What Are The Tools Used In Debugging?**

We generally use logs to see the cause of the issue, where the failure is occurring. So for iOS – **iPhone configuration utility** & for Android **Monitor.bat,**etc can be used. If you provide logs from these tools to the developer, they can easily understand the cause of the issue.

**Question 44: Explain What Does Mobile Security Testing Includes.**

Mobile security testing includes

**Question 45: When To Choose Automation Testing And When To Choose Manual Testing?**

**Manual Testing**

* If the application has new functionality
* If the application requires testing once or twice

**Automate Testing**

* If the regression tests are repeated
* Testing app for complex scenarios

Check out [Manual testing online course](https://www.edureka.co/software-testing) and learn from the expert.

**Question 46: Mention Few Disadvantages of Automation Testing.**

Designing the tools and tests to run software takes a lot of manual, human effort, though there are frameworks and tests ready-made for engineers to use. Even with automated testing, human error is still a factor – tools can be buggy, inefficient, costly, and sometimes even technologically limited in what kinds of tests they can run on their own.

**Question 47.: What Are The Pre-requisites To Start Automation Testing?**

The first step is to segregate the different test cases that are to be automated. Following this, you must prepare test data as per the needs of the test cases. Reusable functions need to be written which are frequently used in those test cases. Later test scripts are prepared by using reusable functions and applying loops and conditions wherever necessary.

**Question 48: What Are The Differences Between Open Source Tools, Vendor Tools, And In-house Tools?**

Open source tools are free to use frameworks and applications. Engineers build the tool and have the source code available for free on the internet for other engineers to use.

Vendor tools are developed by companies that come with licenses to use, and often cost money. Since they are developed by an outside source, technical support is often available for use. Example vendor tools include WinRunner, SilkTest, Rational Robot, QA Director, QTP, LR, QC, RFT, and RPT.

An In-house tool is a tool that a company builds for its own use, rather than purchasing vendor tools or using open source tools.

**Question 49. Is Automation Testing A Complete Replacement For Manual Software Testing?**

No. Proper automation requires as little intervention from humans as possible since the tools used are built to run tests once they are set up. As convenient as this might be, it should not be a complete replacement for manual testing – only for repetitive tasks like [load testing](https://www.edureka.co/blog/load-testing-using-jmeter/), where thousands of virtual users are required.  Engineers should not automate things like test scripts if those scripts can only be expected to run occasionally, nor should they automate code reviews, or bug testing for new builds of software that might require human interaction to detect issues. We can conclude by saying that large-scale, repetitive tasks are a better fit for automation.

**Question 50. What Are The Points That Are Covered In The Planning Phase Of Automation?**

During the planning phase of automation, the points to be considered are as follows:

* Selection of the “right” Automation tool
* Selection of Automation Framework if required
* List of in scope and out of scope items for automation
* Test Environment Setup
* Preparing the Gantt Chart of Project timelines for test script development & execution.
* Identify Test Deliverables