1. **Why Selenium tool**

* **Open-Source:** Selenium is a freeware and a portable tool. It has no upfront direct costs involved. The tool can be freely downloaded and the community-based support for it is freely available.
* **Language support:** Selenium supports a range of languages, including Java, Python, C#, Ruby, JavaScript, etc .
* **Supports Operating Systems:** Selenium can operate and support across multiple Operating Systems (OS) like Windows, Mac, Linux, and UNIX.
* **Support across browsers:** Selenium provides support across multiple browsers like, Internet Explorer, Chrome, Firefox, Opera, and Safari.
* **Tests across devices**: Selenium Test Automation can be implemented for mobile web application automation on Android, iPhone, and Blackberry.
* **Constant updates: Selenium upgrades** are readily available and do not require specific training. This makes Selenium resourceful and cost-effective as well.
* **Ease of implementation:** Selenium offers a user-friendly interface that helps create and execute tests easily and effectively.

1. **Why Choose C#**

* C# is simple yet a modern programming language which is object oriented. One of the main reason for choosing c# is that I have good hands on in C# plus due to its below features.
* C# has features like Properties and Indexers. These features are not available in the Java language.
* C# supports Structures, Operator Overloading and Preprocessors directives, whereas, Java has none of them.
* Through C# we can easily call Windows API function and access COM components which is quite difficult in Java.
* C# features are like
* Type safe
* Interoperability
* Scalable and Updateable
* Component oriented
* Structured programming language
* Rich Library
* Fast speed
* Many more

1. **Why POM Used**

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

For e.g. suppose that there is a small script for a webpage in starting, but with the time test suite grows and as you add more and more lines to the code, things become tough.

The chief problem with script maintenance is that if 10 different scripts are using the same page element, with any change in that element, you need to change all 10 scripts. This is time consuming and error prone.

This is where POM comes in picture which offers a better approach to script maintenance to create a separate class file which would find web elements, fill them or verify them. This class can be reused in all the scripts using that element. In future, if there is a change in the web element, we need to make the change in just 1 class file and not 10 different scripts. This is the reason why POM is widely used and why I have chosen POM.

.