

Stage 1 - Java

Stage 2 - Web automation (Selenium) & AutoIT (Windows GUI automation)

Stage 3 - Framework 1 (TestNG, Data Driven Framework, Page Object Model, Keyword Driven Framework)

Stage 4 - BDD Framework

Stage 5 - Git and Jenkins

Selenium

1. Automates only the web application
2. Open Source
3. Language Independency - Java, C#, Python, Ruby, Javascript

Selenium - A suite of tools

1. Selenium IDE
 - a. No Programming language knowledge required
 - b. Record and playback
 - c. Plugin - chrome, edge, firefox
 - d. Used only simple scripting or exploratory testing
2. Selenium RC - **Deprecated**
 - a. Programming knowledge is required
 - b. Architecture

Source code (Java+Selenium RC jar) → RC server (Turn ON/OFF) → Browser

3. Selenium WebDriver

- a. Programming knowledge is required
- b. Architecture

Source code (Java+Selenium WebDriver jar) → Browser

4. Selenium Grid

If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point

Java Programming

1. Installation

- a. JDK - Java Development Kit (above 11)
 - i. It also installs the JRE

<https://builds.openlogic.com/downloadJDK/openlogic-openjdk/17.0.16+8/openlogic-openjdk-17.0.16+8-windows-x64.zip>

2. IDE

- a. Eclipse

b. IntelliJ

3. Architecture

Source code (.java) → Byte code (.class) → O/P

Compiler → helps to convert source to byte code

JVM → Helps the operating system to understand the byte code

Compile time ⇒ Source code to byte code

Runtime → Bytecode to O/P

4. UpperCamelCase → MyFirstProject

lowerCamelCase → myFirstProject

5. Structure of an eclipse

Workspace

Project - UpperCamelCase

Package - lowercase (com.companyname.purpose)

Class - UpperCamelCase

Methods/variables - lowerCamelCase

6. Datatypes

a. Primitive/Pre-defined datatypes

b. Non- Primitive/Non-predefined/user defined datatypes

i. String

ii. Array

iii. User defined datatypes

7. When we give a whole number, it will be considered as an int

When we store a decimal value, it will be considered as double

8. Debugging

a. Resume

b. Terminate

c. Step over

d. Step into

9. Conditional statement

a. If

b. Switch

10. Iterative statements

a. For

- b. Advance for loop
- c. While
- d. Do while

11. Methods - Building block of the program

Reusability

Maintenance

- a. Static methods
 - i. How to create a static method and call it?
`//accessmodifier static returntype methodName(arguments)`
 To call it
`classname.methodName()`
- b. Non-static methods
 - i. How to create non-static method and call it?
 - ii. How to call non-static method?
 - 1. Create object
 - 2. Use objref.methodName()

12. Variable

- a. Static variable
- b. Non-static variable

13. Object

- a. Declaration
- b. Instantiation - new
- c. Initialization

14. Class - A class is a blueprint or type or template from which objects are created

15. Object

- a. An object is an instance of class
- b. Every object has its own state (non-static variable) and behaviour (non-static method)

16. Access modifier

- a. Private - accessible within the class
- b. Default - accessible within the package
- c. Protected - accessible within the package and also to the inherited classes
- d. Public - accessible anywhere

17. Constructors - Prerequisite of the object

- a. Constructor name and class name should be same without any return type
- b. It is kind of method and it gets called whenever new objects is created
- c. There will be always a default constructor and it helps to load all the non-static variable with default values.
- d. We can override the default constructor by creating explicitly our own constructor

- i. With argument
 - ii. Without argument
 - e. If the class contains constructor with argument then we need to call that only during object creation
18. this
- a. Helps to distinguish between non-static variable and then local variable
 - b. this - will point to current object
19. Constructor overloading / Compile time polymorphism / Static polymorphism
- The constructor to be called is resolved during the compile time
- Can create multiple constructor by change in
- 1. Number of arguments
 - 2. Sequence of arguments
 - 3. Datatype of arguments
20. Method overloading / Compile time polymorphism / Static polymorphism
- The method to be called is resolved during compile time
- Can create multiple method with same name by change in
- 1. Number of arguments
 - 2. Sequence of arguments
 - 3. Datatype of arguments
21. Collections
- a. Non-generic collections
 - i. List
 - ii. HashMap
 - b. Generic collections
 - i. List
 - ii. HashMap

Selenium WebDriver

1. Create Java project
2. Configure the Selenium jar
3. Launch browser
4. Navigate to url, getTitle, getCurrentUrl, getPageSource
5. Click, type, Select
6. To inspect the element → tagname, attribute, text, or not
7. Basic locator

- a. id
- b. name
- c. className
- d. tagName
- e. linkText
- f. partialLinkText

findElement → when there are duplicate locators then findElement picks the first webelement

8. Advance locators

- a. XPath
- b. CSS

9. Synchronization

- a. Unconditional wait (from java)
 - i. Thread.sleep(5000); → not recommended
- b. Conditional wait
 - i. Implicit wait
 - 1. Default Implicit wait - 0s
 - 2. Applicable for all **findElement** and **findElements** methods
 - 3. Example: Implicit wait - 30s
 - a. If the element is not present, it will check for 30 seconds and then throw an exception
 - b. If the element is present, it will do the operation immediately.
 - c. Polling time - 0.5s
 - ii. Explicit wait
 - 1. Exact condition
 - 2. Polling time - 0.5s
 - iii. Fluent wait

10. Dropdown

- a. With select tag
 - i. selectByVisibleText()
 - ii. selectByValue()
 - iii. selectByIndex() → starts at 0
- b. Without select tag

11. Click → element should be present and visible

12. Frame, multiple tabs/windows, alert - switchTo()

13. Multiple tabs/windows

14. List vs Set

- a. List - can contain duplicates
- b. Set - cannot contain duplicates

15. Close vs quit

- a. Close - close the current tab/session

- b. Quit - close the current browser/all session and also it kills the process associated with it
- 16. Frame - embed one html into another html
 - a. Even though the locator is correct, we get **Exception in thread "main"**
org.openqa.selenium.NoSuchElementException:
 - b. Check for frame or iframe tag
 - c. Switch to frame
 - i. Using index
 - ii. Using name or id as a string
 - iii. Using WebElement

17. Alert

- a. Javascript alert

18. Upload

19. Actions

20. Javascripts

21. Chromeoptions

Git - Git is a free and open source distributed version control system

Architecture

Project (in local machine) → local repository (in local machine) → remote repository (github, aws code commit, gitlab, bitbucket)

Concepts →

Modified → staging → commit

Git concepts to update code to remote repo (github or gitlab)

1. `git init` → initialize local repo
2. `git add .` → staging
3. `git commit -m "first commit"` → update the local repo
4. `git remote add origin1`
<https://github.com/balaji-githubstore/selenium-java-concept-equiniti-sep-2025.git> → registering the remote url with name origin1
5. `git push -u origin master`

1 GB - 1024 MB
1 MB - 1024 KB
1 KB - 1024 B
1 B - 8 bits

Exceptions:

1. [NoSuchDriverException](#):
2. [NoSuchElementException](#):
3. [Exception in thread "main"](#)
[org.openqa.selenium.ElementClickInterceptedException](#):- target element is hidden by some other element
4. [ElementNotInteractableException](#) → element is present and not visible
5. [NoSuchWindowException](#)
6. [org.openqa.selenium.NoAlertPresentException](#): no such alert