# 

# **ATB Full Course Curriculum (3-4 Months)**

# 

### **Software Testing Basics (Refresher)**

* Roadmap to Become Manual Tester, Mindset, Salary Discussion & Handle Career Gap & Criteria for Software tester
* What is Software & Its Types, What is Software Testing? , QA Vs QE
* Master Software Development Life Cycle (+STLC)
* All about Black Box Testing & 7+ Techniques
* Types of Software Testing Explain
* Test Planning ( with Test Plan Template) & Defect Reporting ( Bugzilla, JIRA)
* Test Data Management , Understanding Test Execution & Environments
* Requirement Traceability Matrix (with Template), Test Management Tools like TestLink, TMT
* Design your Test cases( Test cases Templates) , Beautiful Test Reporting (Templates)
* Understanding Requirement’s Analysis & Test Plan(A to Z)
* Test Cases, Execution, Test Reporting( with LIVE Project) , Root Cause Analysis (Production Bug Handling)
* LIVE Project Manually Tested x2
* Understanding Agile & Scrum, Scrum PSM 1 Topics and Details
* JIRA Essential for Manual Testers
* Manual Testing Interview Preparation & QnA Discussion, Professional QA Resume
* Mobile Testing Essentials
* How to Test APIs manually

### **API Testing with POSTMAN**

* Client-Server Architecture
* Presentation, Business and Database layers
* HTTP Basics
* HTTP Methods
* What is an API?
* What is API Testing?
* Difference between API testing and Unit testing
* What is Web services?
* HTTP Structure
* Type of Defects in API Testing
* Learn What is xml and JSON?
* URI and URL
* Tools for API Testing
* Introduction to REST Web services.
* SOAP Introduction
* Why learning REST is Important?
* Difference between SOAP and REST Web services.
* REST constraints
* Introduction to POSTMAN tool and installation.
* Creating requests in Postman.
* Creating collections in Postman.
* Understanding variables and environments in Postman.
* Authorization in POSTMAN
* Assertions/Test scripts in POSTMAN.
* Command line integration with Newman.
* Running POSTMAN Collection on Docker, Jenkins
* What is JSON Schema Validation?
* How to perform JSON Schema validation?
* Data Driven Testing with CSV with POSTMAN
* Exploring POSTMAN More features Mocking, Stabbing
* Testing GraphQL with POSTMAN
* Testing Websoctks with POSTMAN
* Writing API Test cases & Test Plan
* Running Testcases with POSTMAN
* Writing API Test cases & Test Plan
* Running Testcases with POSTMAN
* Advance JSON Schema
* **LIVE API Testing Example**

### **Core Java for Software Testers**

* What is Java, Features, Variables?
* Data Types in Java- Primitive Data Type and Non-Primitive Data Type
* Operators-Arithmetic and Logical operators in Java
* Looping statement- For loop, while loop, enhanced for loop
* Conditional Statement
* Array in Detail
* Classes, Object in Java
* Static vs Non-static
* Access Modifier- Public, Private, Protracted, Default
* Inheritance- Single level Inheritance
* Constructor- Default constructor and parameterized constructor
* Method overloading- Compile time Polymorphism
* Constructor overloading- Compile time Polymorphism
* Method overriding- Run time Polymorphism
* Interface
* Abstract Class
* String class and usage
* Collection and Collections- Dynamic data structure and important for the framework and script development
* List- ArrayList and LinkedList
* Set- HashSet, LinkedHashSet,TreeSet
* Map- HashMap, LinkedHashMap, TreeMap
* Exception Handling- Handle Java exception
* Best Practices Programming & Framework

### **Maven Mastery for API Automation Project**

* What is Maven and Why Maven Required for Automation
* Download and Installing Maven in Windows and plugin in eclipse
* Creating first build in Maven
* Different Goals in Maven
* How to parameterized Maven build
* Running maven build from eclipse and from the command prompt

### **Learn TestNG for API Automation**

* TestNG-UnitTest Framework
* TestNG in Details
* Advantage of using TestNG
* Download and Write the first test script in TestNG
* TestNG document understanding
* Different annotation usage
* How to run the test in sequence manner
* How to provide dependency between test cases
* How to use assert values and validation
* Soft Assert vs Hard Assert
* Reports in TestNG
* Creating a test suite in TestNG to run multiple test cases
* Passing parameter to test via testng.xml file
* Logs in TestNG

### **API Automation with Rest Assured**

* Configure Eclipse with Rest-Assured
* REST API Test using Rest Assured
* Validate Response Status using Rest Assured
* Validate Response Header using Rest Assured
* Read JSON Response Body using Rest Assured
* Framework Definitions
* Type of Automation framework, usage, features and myth about the framework
* Data Driven Framework from scratch
* Hybrid Framework
* Creation of framework Skelton
* Different component of the framework
* Data Provider- Config and Excel Data Provider- Test Data and config file
* Base Class creation and usage
* Folder structure & Framework creation
* Framework creation
* Framework creation
* Setting up Cucumber project
* Performing a LIVE project Cucumber BDD and Rest Assured
* Running API Automation in CI/CD Jenkins
* Discussion over more than 51+ interview questions for freshers and experience candidate

### **Mastering Web Automation with Selenium**

* What is Selenium
* Why Selenium? / Advantages
* What are its versions?
* What all OS, Browsers, and Programming Languages does it Support?
* Java-Selenium Architecture
* Selenium vs Playwright vs Cypress
* WebDriver Architecture
* Selenium IDE
* Selenium Grid 3, 4
* Basic Selenium Program to Open and close Browser
* Runtime Polymorphism Program in Selenium
* WebDriver abstract methods
* Locators XPath, its Types and cases.
* Handling Multiple Elements
* Handling Synchronization issue by using implicitly Wait and Explicit Wait
* Handling Dropdown (static and dynamic)
* Handling Keyboard and Mouse Actions
* Taking Screenshot
* Handling Disabled Element , Performing Scroll down Action
* Web Element Interface Methods
* Handling Popups (web-based and Window-based)
* Handling Frames, Handling New Windows/New Tabs
* Custom Automation Framework
* New Selenium 4.16
* Cloud Testing - BrowserStack Demo

### **Custom Test Automation Framework UI + API**

* Advance Selenium & Selenium Grid
* Own Serenity BDD Selenium Framework.

### **Mastering SQL**

What Is Database?

Database Management System (DBMS)?

What is Relational Model ?

Datatypes and Constraints

• What are Datatypes ?

• Types and Examples .

• How to use .

• What are Constraints?

• Types and Examples.

• How to use.

Statements in SQL

• Data Definition Language (DDL)

• Data Manipulation Language (DML)

• Transaction Control Language (TCL)

• Data Control Language (DCL)

• Data Query Language (DQL)

Software installation

• Installing and set up of software

• Working on Oracle 10g.

Data Query Language (DQL)

• Select

• From

• Where

• Group By

• Having

• Order By

Operators

• Types and Examples

Functions in SQL

• Single Row Functions

• Multi Row Functions

Max ()Min () , Sum (), Avg () , Count ()

Sub Query

JOINS

• What Is Join?

• Types of Joins.

• Cartesian Join

• Inner Join

• Outer Join

• Self-Join

• Queries and Examples.

Data Definition Language (DDL)

• Create

• Rename

• Alter

• Truncate

• Drop

Data Manipulation Language (DML)

• Insert

• Update

• Delete

Transaction Control Language (TCL)

• Commit

• Save point

• Rollback

Data Control Language (DCL)

• Grant

• Revoke

Normalization

• Introduction to Normalization

• Types of Normal Forms

• Examples.

### **Mobile Testing and Appium Overview**

* What is Mobile Testing?
* Appium Installation Instructions for Windows & Mac
* Getting Started with Mobile testing using Appium Code
* Appium Concepts
* R
* eal time Project X1

### **Resume Prep, Automation Testing Interview QnA & Mock Interview**

* Fix your Resume.
* Mock Interviews
* LinkedIn hacks to Get IT Calls

### **Doubt Clearing session, Interview Prep & Discussion**

* 2 Sessions with Code review, Assignment review

[**Enroll Now**](https://courses.thetestingacademy.com/courses/job-ready-automation-tester-blueprint-with-java-by-pramod-dutta/buy)

## **Notes - TheTestingAcademy (Pramod Sir)**

## **( Demo : Ref Notes) Full notes will be shared in Class**

### **Mastering Web Automation with Selenium**

#### Run your First Selenium Script

1. Install Java and Set path to the Home in Windows / Mac
2. Install IDE (IntelliJ) and you can avoid the first step.
3. Create a New Quick Start Maven Project, Add dependencies.
4. Download Browser drivers ( can skip if Selenium 4.6.0).
5. Add the code and run the program.

#### How to Set JAVA\_HOME path in MAC - [Click here](https://stackoverflow.com/questions/22842743/how-to-set-java-home-environment-variable-on-mac-os-x-10-9)

1. Download the JDK latest -

| echo export "JAVA\_HOME=\$(/usr/libexec/java\_home)" >> ~/.bash\_profile |
| --- |

1. If you're using zsh (which probably means you're running macOS Catalina or newer), then it should instead be:

| echo export "JAVA\_HOME=\$(/usr/libexec/java\_home)" >> ~/.zshrc |
| --- |

1. In either case, restart your shell.

#### **How to Set JAVA\_HOME path in Widnows**

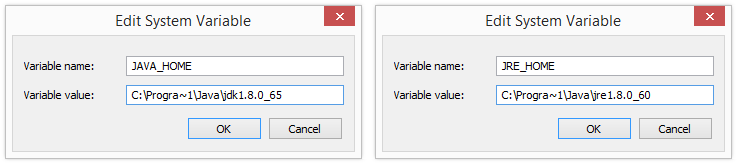
Set the JAVA\_HOME Variable

To set the JRE\_HOME or JAVA\_HOME variable:

1. Locate your Java installation directory  
   If you didn't change the path during installation, it'll be something like

| C:\Program Files\Java\jdk1.8.0\_65 |
| --- |

You can also type where java at the command prompt.

1. Do one of the following:  
   **Windows 7** – Right click **My Computer** and select **Properties** > **Advanced  
   Windows 8** – Go to **Control Panel** > **System** > **Advanced System Settings  
   Windows 10** – Search for **Environment Variables** then select **Edit the system environment variables**
2. Click the **Environment Variables** button.
3. Under **System Variables**, click **New**.
4. In the **Variable Name** field, enter either:
   * JAVA\_HOME if you installed the JDK (Java Development Kit)  
     or
   * JRE\_HOME if you installed the JRE (Java Runtime Environment)
5. In the **Variable Value** field, enter your JDK or JRE installation path .  
   If the path contains spaces, use the shortened path name. For example, C:\Progra~1\Java\jdk1.8.0\_65  
     
      
   Note for Windows users on 64-bit systems  
   Progra~1 = 'Program Files'  
   Progra~2 = 'Program Files(x86)'
6. Click **OK** and **Apply Changes** as prompted

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### Browser driver classes and Webdriver Interface

| *// Import the necessary modules* **import** org.openqa.selenium.WebDriver; **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** **TestingAcademy** {   **public** **static** **void** **main**(String[] args) {  *// Set the path to the Chrome driver*  System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");   *// Create a new Chrome browser instance*  WebDriver driver = **new** ChromeDriver();   *// Navigate to the Testing Academy website*  driver.get("http://thetestingacademy.com/");  } } |
| --- |

WebDriver IS Class OR interface.

WebDriver is an interface that defines a set of methods for interacting with web pages. It is not a class, but rather a collection of abstract methods that must be implemented by classes that implement the WebDriver interface.

Classes that implement the WebDriver interface, such as the ChromeDriver class, must provide implementations for these methods in order to use the WebDriver interface.

The **ChromeDriver** class provides a number of methods for interacting with the Chrome browser, such as get() for navigating to a specific URL, findElement() for locating elements on a page, and click() for simulating a mouse click on an element. You can use these methods to automate a variety of actions on the Chrome browser.

…….

Demo…..

Write a program to verify the current title and URL in Selenium?

| *// Import the necessary modules* **import** org.openqa.selenium.WebDriver; **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** **VerifyTitleAndUrl** {   **public** **static** **void** **main**(String[] args) {  *// Set the path to the Chrome driver*  System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");   *// Create a new Chrome browser instance*  WebDriver driver = **new** ChromeDriver();   *// Navigate to the specified URL*  driver.get("https://www.google.com");   *// Get the current page title and URL*  String title = driver.getTitle();  String url = driver.getCurrentUrl();   *// Print the title and URL to the console*  System.out.println("Title: " + title);  System.out.println("URL: " + url);   *// Close the browser*  driver.close();  } } |
| --- |

| *// Set the paths to the Edge and Firefox drivers*  System.setProperty("webdriver.edge.driver", "/path/to/msedgedriver");  System.setProperty("webdriver.gecko.driver", "/path/to/geckodriver"); |
| --- |

#### ChromeOptions

ChromeOptions is a class in the Selenium Python bindings that allows you to create and configure Chrome browser instances with various options. This can be useful when you want to customize the behavior of the browser or work around issues with the default settings.

#### 

#### 

#### 

#### Locators in Selenium

They are used to identifying and locate elements on a webpage so that they can be interacted with. There are different types of locators in Selenium, including:

* ID: locates elements with a specific ID attribute value
* Name: locates elements with a specific name attribute value
* Class name: locates elements with a specific class attribute value
* Tag name: locates elements with a specific HTML tag name
* Link text: locates anchor elements with specific text
* Partial link text: locates anchor elements with partial text
* CSS selector: locates elements matching a CSS selector
* XPath: locates elements matching an XPath expression

…….Full Notes in Class



Enroll now…

<https://sdet.live/become>