TESTING SYLLABUS

MODULE 1: SOFTWARE TESTING CONCEPTS

- Software Testing Introduction
- Software Development Life Cycle (SDLC)
- Static Testing, Reviews and Walkthroughs
- White Box Testing
- Unit Testing & Integration Testing
- Black Box Testing
- System Testing & User Acceptance Testing
- Alpha Testing & Beta Testing
- Functional Testing Types
- Smoke Testing / Sanity Testing
- Formal Testing
- Ad-hoc Testing
- Re-Testing
- Regression Testing
- System Integration Testing
- End-to-End Testing
- Exploratory Testing
- Monkey Testing
- Non-Functional Testing Types
- UI Testing
- Usability Testing
- Security Testing
- Compatibility Testing
- Load Testing
- Performance Testing
- Globalization Testing
- Localization Testing
- Recovery Testing

MODULE 2: MANUAL TESTING OR STLC

- Software Testing Life Cycle (STLC)
- What is STLC?
- STLC Phases
- Test Plan
- Test Analysis
- Analysing Functional Requirements SRS/FRS
- Test Design
- Preparing Test Scenarios
- Preparing Test Cases
- Preparing Test Data
- RTM
- Test Case Execution
- Defect Reporting
- Defect Severity & Defect Priority
- Defect / Bug Life Cycle
- Defect Management
- Test Closure

MODULE 3: AGILE METHODOLOGY USING SCRUM

- What is Agile Testing?
- Scrum Introduction
- Getting Agile with Scrum
- Scrum Approach
- Release Planning
- Sprint Planning
- Product backlog
- Concept of Epics
- Writing Epics – Examples

- Concept of User Stories
- Writing User Stories- Examples
- Defining Tasks
- Starting Sprint
- Monitoring Sprint status
- Completing Sprint

MODULE 4: CUCUMBER STUDIO (TEST MANAGEMENT TOOL)

- What is Behaviour-Driven Development (BDD)?
- Advantages of BDD in Agile development
- Overview of Cucumber & Gherkin syntax
- Creating a Smart Bear account
- Navigating the Cucumber Studio interface
- Creating and organizing projects
- Understanding test cases and folders
- Test run reports
- Living documentation and traceability
- Understanding status of features and scenarios

MODULE 5: JIRA (PROJECT MANAGEMENT TOOL)

- JIRA Introduction
- How to Install JIRA
- JIRA Features
- Creating Scrum Project in JIRA
- Adding Users to our JIRA Account
- JIRA Issue types
- Workflow for a Project
- Creating Product backlog in JIRA
- Creating EPICS in JIRA
- Creating User Stories in JIRA
- Starting Sprint
- Writing Testcases in JIRA
- Executing Testcases from JIRA
- Adding Bugs to the JIRA Project
- Attaching screenshots of defects in JIRA

MODULE 6: SQL FOR DATABASE TESTING

- What is Database Testing?
- Why is DB Testing Necessary?
- SQL Overview
- SQL CREATE Database
- SQL CREATE Table
- SQL SELECT Query
- SQL WHERE Clause
- SQL SELECT DISTINCT
- SQL GROUP BY
- SQL HAVING
- SQL ORDER BY Keyword
- SQL LIKE Operator
- SQL IN Operator
- SQL BETWEEN Operator
- SQL AND & OR Operators
- SQL Aggregate Functions
- SQL INSERT
- SQL UPDATE
- SQL DELETE
- SQL TRUNCATE
- SQL ALTER
- SQL DROP
- SQL JOINS
- SQL SUB QUERIES

AUTOMATION TESTING MODULES

MODULE 7: AUTOMATION TESTING OVERVIEW

- What is Automation Testing?
- Why Testing Tools?
- Manual vs Automation Testing
- Advantages of Automation Testing
- When to Start Automation Testing?
- Tool Evaluation & POC
- Types of Automation Tools (Licensed & Open Source)
- Test Automation Fundamentals
- Test Automation Tool Evaluation
- Introduction to Selenium
- Difference Between Selenium & QTP
- Selenium Components: IDE, RC, WebDriver, Grid

MODULE 8: CORE JAVA FOR AUTOMATION

- Core Java Fundamentals
- About Eclipse
- Installation of Java
- Installation of Eclipse IDE
- Creating Project in Eclipse
- Writing & Executing Programs in Eclipse IDE
- Eclipse and Selenium Together
- Debugging using Eclipse
- History & Features of Java
- Introduction to Java Programming
- Keywords & Class and Object
- Data Types & Variables
- Arrays, Construction, and Initialization
- Conditional Statements (if-else, nested if-else, switch case)
- Loop Statements (while, do-while, for, foreach)
- Keywords (break, continue)
- What is an Array & Advantages of Arrays in Java
- Working with Arrays & For Each Loop
- Examples for Arrays
- What are Packages?
- Creating Packages, Class & Objects
- Types of Methods
- Writing User-Defined Methods
- Inheritance (What is Inheritance, Advantages, Types, Examples)
- Polymorphism (Method Overloading & Method Overriding, Examples)
- Encapsulation (Introduction, Examples)
- Abstraction (Define Abstract Class & Methods, Steps to Define Abstraction)
- Interface (Implementing Interface, Difference between Abstraction & Interface, Implements vs Extends)

MODULE 9: SELENIUM WEBDRIVER WITH JAVA

- Introduction to Selenium
- Advantages of Selenium WebDriver
- Selenium WebDriver vs Selenium RC
- Selenium WebDriver Architecture
- Configuring Selenium with Eclipse (Adding JAR Files)

- Working with WebDriver on Different Browsers (Firefox, Chrome, IE, Edge)
- Creating First WebDriver Script
- Navigate Methods in WebDriver
- Locators in Selenium (ID, Name, ClassName, TagName, LinkText, PartialLinkText, CSS, XPath)
- XPath & XPath Axes (Self, Child, Parent, Following, Preceding, Ancestor, Descendant)
- Static & Dynamic Dropdowns
- Handling Checkboxes & Radio Buttons
- Synchronization & Wait Methods (Thread.sleep, Implicit, Explicit, Fluent Wait)
- Handling Alerts
- Handling Frames & Iframes
- Handling Windows & Popups (WindowHandles, Switching Windows)
- Working with WebTables
- JavaScriptExecutor in Selenium
- Capturing Screenshots on Test Failure
- Handling AJAX Events & Auto Suggestions
- Handling Different Selenium Exceptions
- File Upload & Download using Selenium
- Working with Apache POI (Reading & Writing Excel Data in Selenium)
- Using Robot Class for Keyboard & Mouse Events
- Handling Windows Popups using Robot Class
- Mouse Events using Action Class (Mouse Hover, Drag & Drop, Double Click, Right Click)

MODULE 10: TESTNG FRAMEWORK

- Introduction to TestNG
- Advantages of TestNG over JUnit
- TestNG Features
- Configuring TestNG with Eclipse
- Writing Selenium Testing Scripts using TestNG
- Running TestNG Tests & Checking Reports
- TestNG Annotations (@Test, @BeforeMethod, @AfterMethod, @BeforeClass, @AfterClass, @BeforeTest, @AfterTest, @BeforeSuite, @AfterSuite)
- TestNG Tags & Attributes (enabled, dependsOnMethods, priority)
- Assertions in TestNG (Soft Assertions & Hard Assertions)
- Data Providers & Parameterization using @DataProvider
- TestNG Listeners (Implementing ITestListener, ISuiteListener)
- Generating TestNG Reports (HTML, XML)
- Creating and Running Test Suites using TestNG (Testng.xml)
- Skipping Tests in TestNG
- Setting Priority of Test Execution
- Executing Regression, Sanity, Smoke Test Cases in TestNG
- Grouping Tests & Running Test Groups
- Parallel Test Execution & Configuring Parallel Execution in TestNG XML
- Dependency Tests (dependsOnMethods, dependsOnGroups)
- Execution Order & Thread Safety in Parallel Execution
- Advanced TestNG Concepts (Printing Log Statements in TestNG Report, TestNG Results Output Folder Walkthrough, Running Parallel & Distributed Tests)

MODULE 11: CUCUMBER & BDD FRAMEWORK

- Introduction to Cucumber
- What is Cucumber?
- Introduction to Behaviour-Driven Development (BDD)
- Why use Cucumber for Automation Testing?
- Comparison with Traditional Testing Frameworks (JUnit, TestNG)
- Benefits of BDD in Agile Environments
- Difference Between Cucumber & Other Automation Tools
- Setting Up Cucumber
- Installing Cucumber in Eclipse/IntelliJ
- Setting up Maven/Gradle with Cucumber Dependencies
- Gherkin Syntax & Writing Feature Files
- What is Gherkin?
- Role of Gherkin in BDD
- Gherkin Syntax & Structure
- Writing Feature Files: Feature, Scenario, Given, When, Then
- Gherkin Keywords (And, But, Scenario Outline, Examples)
- Data Tables & Comments in Gherkin
- Handling Complex Data with Tables
- Step Definitions in Cucumber
- What are Step Definitions?
- Mapping Gherkin Steps to Java Code
- Writing Step Definitions (@Given, @When, @Then)
- Regular Expressions in Step Definitions
- Passing Parameters & Data Between Steps
- Hooks in Cucumber (@Before, @After Annotations for Setup & Teardown)
- Cucumber with Selenium WebDriver
- Integrating Cucumber with WebDriver
- Preparing Feature Files for Test Scenarios
- Writing Runner Class in Cucumber
- Usage of Given, When, Then, And, But in Feature Files

MODULE 12: GIT & VERSION CONTROL

- Starting a Git Repository
  - git init – Creating a new Git repository
- Checking Git Status
  - git status – Checking the state of your files (modified, staged, untracked)
- Staging Files
  - git add <file> – Adding files to the staging area
  - git add . – Staging all files
- Committing Changes
  - git commit -m "Commit Message" – Creating a commit with a message
  - Writing good commit messages
- Viewing Commit History
  - git log – Viewing commit history

MODULE 13: FRAMEWORK EXPLANATION

- Page Object Model (POM)
- Page Factory
- Folder Structure in Automation Framework
- Utility Class in Framework Development
- Report Creation in Selenium Framework
- Data driven and key word driven framework

MODULE 14: MAVEN & BUILD MANAGEMENT

- What is Maven?
- Using pom.xml File
- Setting Up Maven Project in Eclipse
- Building & Running Selenium Maven Project
- Adding Dependencies in pom.xml
- Updating Selenium Project Dependencies using Maven

MODULE 15: INTEGRATION OF THIRD-PARTY TOOLS

- Log4j
  - Log4j Introduction
  - Usage of Logger
  - Adding Log4j Jars
  - Integrating Log4j in Selenium
- Jenkins CI/CD
  - What is Jenkins?
  - Installing Jenkins
  - Steps to Integrate Jenkins with Selenium
  - Configuring Email Notifications in Jenkins
  - Scheduling Jenkins Jobs
  - Running Selenium Scripts via Jenkins
  - Creating & Executing Batch Files

MODULE 16: TESTING OF WEBSERVICES/API

- Introduction of API and Web services:
  - What is API
  - What is Webservice
  - What is Backend Architecture
  - Rest vs SOAP APIs
  - What is CRUD operation with different examples
  - What are different HTTP Calls – GET/POST/PUT/Delete/PATCH/HEAD/ OPTIONS
  - Different Live Projects Examples
- Postman:
  - Introduction
  - How to call Rest API in Postman
  - How to pass path parameter in Request
  - How to pass query parameter in Request
  - How to set Headers in Postman
  - How to pass JSON/XML Payload
  - How to check response status code
  - How to check JSON/XML response messages
  - What is response header
  - Header Presets
  - API History in Postman
  - Postman Environment Variables
  - Run Automated Test Cases in Postman and generate the results
- Advance Postman/Newman:
  - Introduction of Newman
  - Newman installation/setup
  - How to run collections from Newman
  - Newman command line options
  - Generate HTML Test Report
  - Create a collection and folders
  - Create a collection from API Specification or API Schema
  - maintenance of API collections
  - Postman Console and logs
  - Test Script in postman
  - Pre Request Script in postman
  - Test Code snippet in postman
  - Monitoring of APIs
  - API Documentation
  - API Documentation/CODES in different programming languages
  - Publish Documentation
  - Setup Environment – Local and Global
  - Collection Runner
  - Collection Test Summary
  - How to create a team workspace
  - How to create individual workspace
  - Import in Postman

- What are different HTTP Status Response Codes:
  - 1xx series
  - 2xx series
  - 3xx series
  - 4xx series
  - 5xx series

MODULE 17: AUTOMATION TESTING OF BACK END/APIS

- Learn various HTTP Client libraries:
  - Tools/Library: JDK 11 – HTTP Client
  - All CRUD – GET/POST/PUT/Delete Calls
  - How to send Request with Payloads
  - How to send Query Parameters and Path Parameters
- Learn Rest Assured DSL Java Client API:
  - Introduction of Rest Assured
  - Rest Assured methods: GET/POST/PUT/Delete/Patch
  - Handle different payloads using POJO and JSON/XMLs:
  - Request with Simple JSON format
  - Request with Complex JSON format
  - Handle Authentication using:
    - Basic Auth
    - Pre-emptive Auth
    - Digestive Auth
    - Bearer Token
    - Authorization Token
    - Client ID
    - Client Secret
    - API Key
    - API Secret
    - OAuth1 and OAuth2
    - JWT Token
    - API KEY

MODULE 18: AUTOMATION ALL CORE FEATURES OF REST ASSURED LIBRARY

- GET/POST/PUT/DELETE – Inbuilt BDD Approaches
- What is BDD GIVEN/WHEN/THEN/AND methods
- What is LOG and ALL methods
- How to ASSERT in Rest Assured
- How to verify Status Code, Response Body and Headers
- How to use QUERY PARAMETERS and PATH PARAMETERS
- What is ResponseBuilderSpec and Request Specification Concept
- What is JSON PATH/XML Path
- How to create a JSON Path to get the data from a complex JSON response using JAYWAY Json Path library.
- How to create a XMLPath to get the data from a complex JSON response using Rest Assured XMLPath library.
- What is EXTRACT method
- How to send a POST CALL with JSON String, JSON File and using Java (POJO) Class.
- What is Serialization and De-Serialization
- How to use HAMCREST – Assertion Library
- GET/POST/PUT/DELETE – NON-BDD Approaches

Real Time API Examples:

- GORest APIs with
- Authentication ReqRes.in
- Booke APIs
- Weather APIs
- GIT HUB APIs
- API with Swagger

MODULE 19: AGENTIC TESTING (NEXT-GEN TESTING)

Agentic Concepts

- What are AI agents?
- Agent workflows in QA
- Autonomous test creation & execution

Agentic Testing Use Cases

- Auto-exploration of application
- Dynamic test generation
- Failure analysis & self-repair

Hands-On Concepts

AI agents for:

- Test case creation
- Bug reproduction
- Test maintenance

Examples

- LLM-based test agents
- Selenium + LLM agents
- Autonomous API testing agents