



# SDET Professional Training Course



# Table of Contents

<b>1. SDLC, Agile &amp; Scrum</b>	<b>4</b>
<b>2. Manual Testing</b>	<b>4</b>
<b>3. Core Java</b>	<b>7</b>
<b>4. TestNG</b>	<b>8</b>
<b>5. Web Automation using Selenium</b>	<b>9</b>
<b>6. Automation Frameworks</b>	<b>10</b>
<b>7. BDD using Cucumber</b>	<b>11</b>
<b>8. API Testing with Postman</b>	<b>12</b>
<b>9. API Test automation using RestAssured</b>	<b>13</b>
<b>10. Version Control, Git &amp; Github</b>	<b>13</b>
<b>11. DevOps &amp; CI/CD</b>	<b>14</b>
<b>12. Database &amp; SQL</b>	<b>14</b>
<b>13. ALM using Jira</b>	<b>15</b>
<b>14. Test Management Tool - Xray</b>	<b>16</b>

## About SDET Course

Welcome to the SDET training course, where we will provide you with the knowledge and skills necessary to become a highly effective Software Development Engineer in Test.

In this comprehensive training program, you will learn how to apply your programming and testing expertise to build and maintain robust, scalable, and efficient software testing frameworks. Our experienced trainers will guide you through the latest tools, technologies, and best practices in the field of software testing, including Agile methodologies, SDLC, Core Java, TestNG, Selenium, Automation Frameworks, Cucumber, API Testing with Postman, API Test Automation using RestAssured, Appium, Version Control, Git & Github, DevOps & CI/CD, Security Testing, Performance Testing, Cross-Browser Testing using BrowserStack, ALM using Jira, and Test Management Tool - Xray.

By the end of the program, you will be equipped with the skills to design and implement effective test automation frameworks, automate testing for web and mobile applications, perform API testing, implement continuous integration and deployment, and execute security and performance testing. Additionally, you will be able to collaborate seamlessly with developers, testers, and stakeholders to ensure the quality and reliability of the software products.

Our training program is designed for both beginners and experienced professionals who want to enhance their skills and knowledge in the field of software testing. Our trainers have years of hands-on experience in the software testing industry and will provide you with practical training that is relevant to the industry.

So, if you are looking to advance your career as an SDET or want to enter the field of software testing, this training program is the perfect opportunity for you. Enroll now and take the first step towards becoming an expert Software Development Engineer in Test.

# Course Content

## 1. SDLC, Agile & Scrum

- a. SDLC Phases
  - i. Requirements Phase
  - ii. Analysis Phase
  - iii. Design phase
  - iv. Coding Phase
  - v. Testing phase
  - vi. Deployment phases
  - vii. Maintenance Phase
- b. SDLC Models
  - i. Waterfall Model
  - ii. Spiral Model
  - iii. Prototype Model
  - iv. V Model
  - v. Hybrid Model
  - vi. Agile Model
- c. What is Agile
- d. Agile Values - Manifesto
- e. Introduction to Scrum
- f. Scrum Framework
- g. Scrum Roles
- h. Scrum Ceremonies and Events
- i. Scrum Terms and Artifacts
- j. Estimation In Agile

## 2. Manual Testing

- a. Why Quality?
- b. Software Characteristics
- c. What is Software Quality?
- d. What is software testing?
- e. Testing Vs Debugging
- f. QA vs QC vs Testing
- g. Seven Testing Principles
- h. Cost of Quality
- i. The Psychology of Testing
- j. Test Process
- k. Test Process in Context
  - i. Test Activities and Tasks
  - ii. Test Work Products

- iii. Traceability
- I. Testing Throughout the Software Development Lifecycle
  - i. Test Levels
    - 1. Component Testing
    - 2. Integration Testing
    - 3. System Testing
    - 4. Acceptance Testing
  - ii. Test Types
    - 1. Functional Testing
    - 2. Non-functional Testing
      - a. Performance Testing
      - b. Compatibility Testing
      - c. Security Testing
      - d. Recovery Testing
      - e. Accessibility Testing
      - f. I18N Testing
      - g. L10N Testing
      - h. Usability Testing
    - 3. Testing Methodologies
      - a. White Box Testing.
      - b. Black Box Testing.
      - c. Grey Box Testing
    - 4. Change-related Testing
      - a. Regression Testing
      - b. Retesting
    - 5. Test Types and Test Levels
  - iii. Maintenance Testing
    - 1. Triggers for Maintenance
    - 2. Impact Analysis for Maintenance
    - 3. Sanity/Smoke Testing
    - 4. Retest/Regression Testing
  - iv. Static Testing
    - 1. Work Products that Can Be Examined by Static Testing
    - 2. Benefits of Static Testing
    - 3. Static Vs Dynamic Testing
    - 4. Review Process
      - a. Work Product Review Process
      - b. Roles and responsibilities in a formal review
      - c. Review Types
      - d. Applying Review Techniques
      - e. Success Factors for Reviews
- m. Test Case Design Techniques
  - i. Introduction to Test Case
  - ii. Categories of Test Techniques
  - iii. Black-box Test Techniques
    - 1. Boundary value analysis

2. Equivalence class partitioning
3. Decision Table Testing
4. State Transition Testing
5. Use Case Testing
- iv. White-box test techniques
  1. Statement Testing and Coverage
  2. Decision Testing and Coverage
  3. The Value of Statement and Decision Testing
- v. Experience-based Test Techniques
  1. Error guessing
  2. Exploratory Testing
  3. Checklist-based Testing
- vi. Test case template and attributes in it
- vii. Writing test cases and practice
- viii. Using mind maps in test case design
- n. Test Management
  - i. Test Organization
  - ii. Independent Testing
  - iii. Tasks of a Test Manager and Tester
  - iv. Test Planning and Estimation
    1. Purpose and Content of a Test Plan
    2. Test Strategy and Test Approach
    3. Entry Criteria and Exit Criteria
    4. Test Execution Schedule
    5. Factors Influencing the Test Effort
    6. Test Estimation Techniques
  - v. Test Monitoring and Control
    1. Metrics Used in Testing
    2. Purposes, Contents, and Audiences for Test Report
  - vi. Configuration Management
  - vii. Risks and Testing
    1. Definition of Risk
    2. Product and Project Risks
    3. Risk-based Testing and Product Quality
  - viii. Defect Management
    1. Introduction to Defects
    2. Defect Life Cycle
    3. Status of Defects
    4. The severity of a defect
    5. Priority of a defect
    6. Defect Report/Incident Report
    7. Defect Tracking Tool - Jira

**[Note:- In Professional Automation Course - Above Manual testing syllabus can be omitted for customization. but everything else remains the same. Feel free to ask if you have questions! ]**

### 3. Core Java

- a. Introduction and Installation
  - i. History of Java
  - ii. Installing the Java Development Kit
  - iii. The basics of the Java programming language
  - iv. Writing and running a simple Java program
- b. Structure of programming language
  - i. Data Types
    - 1. Primitive - Int, Boolean, Char, float, long, double
    - 2. Non-Primitive - String, arrays
  - ii. Operators
  - iii. Keywords
  - iv. Access modifiers
- c. Control Flow Statements
  - i. If – else
  - ii. If – else – if
  - iii. Nested if
  - iv. Switch case
- d. Looping in Java
  - i. For Loop
  - ii. While Loop
  - iii. Do – while Loop
  - iv. Continue statement
  - v. Break statement
  - vi. Inner Loop / Nested Loop
- e. Object-oriented Concepts
  - i. Classes and Objects
  - ii. Object creation
  - iii. Reference variable
  - iv. Global and local variables
  - v. Constructors
  - vi. Aggregation
  - vii. Composition
  - viii. Encapsulation
  - ix. Inheritance
    - 1. Single inheritance
    - 2. Multilevel inheritance
    - 3. Hierarchical inheritance
  - x. Polymorphism

1. Method overloading
    2. Method overriding
  - xi. Abstraction
    1. Abstract class
    2. Interface
  - xii. Variables
    1. Local variable
    2. Instance variable
    3. Static / Global variable
  - xiii. Methods
    1. Declaration
    2. Parameterization
    3. Returning value
    4. Automatic promotion
    5. Method signature
- f. super keyword
- g. this keyword
- h. final keyword
- i. Typecasting
- j. Java packages
- k. Exception Handling
- l. Generics
- m. Collections & streams - ArrayList, Hashmaps, hash tables
- n. Multithreading
- o. File I/O
- p. Lambda Expressions
- q. Functional interfaces
- r. Method References
- s. Default Methods
- t. Optional Class
- u. Serialization
- v. Garbage Collector
- w. New Date/Time API

#### 4. TestNG

- a. Introduction to TestNG
  - i. History and Overview of TestNG
  - ii. Installing TestNG in Eclipse or IntelliJ IDEA
  - iii. Writing and running a simple TestNG test
- b. TestNG Annotations
  - i. Understanding test annotations and attributes
  - ii. Configuring test behavior using annotations
  - iii. Using TestNG assertions
- c. TestNG Test Suites
  - i. Creating test suites in TestNG



- ii. Grouping tests in test suites
- iii. Running test suites using TestNG
- d. TestNG Parameterization
  - i. Parameterizing tests in TestNG
  - ii. Data Providers and Data Sources
  - iii. TestNG Factory annotation
- e. Data-Driven Testing in TestNG
  - i. Data-driven testing concepts and best practices
  - ii. Data-driven testing using TestNG
  - iii. Reading test data from Excel, CSV, or JSON files
- f. Parallel Test Execution in TestNG
  - i. Overview of parallel test execution in TestNG
  - ii. Configuring parallel test execution in TestNG
  - iii. Parallelism at the test and suite level
- g. TestNG Listeners
- h. Reports using TestNG
- i. Final Project
  - i. Students will complete a final project that demonstrates their understanding of TestNG for test automation. The project will involve writing a set of automated tests using TestNG, and one of the integration tools covered in the course. The project will be graded based on functionality, design, and code quality.

## 5. Web Automation using Selenium

- a. Introduction to Test Automation
  - i. What is Automation Testing?
  - ii. When will we switch to Automation Testing?
  - iii. Why Automation testing?/Advantages/Disadvantages
  - iv. Automation Testing Tools
- b. Selenium IDE
- c. Selenium WebDriver Overview
- d. Locating Elements
  - i. id
  - ii. name
  - iii. className
  - iv. tagName
  - v. linkText
  - vi. partialLinkText
  - vii. cssSelector
  - viii. Xpath
    - 1. Absolute
    - 2. Relative X- path
      - a. Contains
      - b. Ancestor/descendants
      - c. And/OR

- d. Parent
- e. Starts With
- f. Preceding/following
- g. Siblings
- e. Handling Multiple Elements
- f. Navigation in selenium-
- g. Handling Edit-box-
- h. Handling Disabled Element
- i. Taking Screenshot
- j. Performing Scroll down Action
- k. Handling Drag and Drop
- l. Handling Keyboard and Mouse Actions
- m. Handling Mouse Hover
- n. Keyword Events using Action class
- o. Handling Popups (web-based and Window-based)
- p. Handling New Windows/New Tabs
- q. Scrolling on a web page using JavaScript Executor
- r. Types of Alerts
  - i. Handling Alerts
  - ii. Handling multiple windows & tabs
  - iii. Verify Page title in Selenium WebDriver
- s. Handling links
- t. Handling Radio button & Check-box
- u. Handling WebTable
- v. Handling Drop Down using Select class
- w. Methods under Select class
- x. Resize operations
- y. Handling File Upload
- z. What is an IFrame
  - i. Identifying an IFrame
  - ii. Switching to a specific IFrame in Selenium WebDriver
- aa. Working with Excels
- bb. Handling Synchronization issues by using implicitlyWait and Explicit Wait

## 6. Automation Frameworks

- a. Stages of Automation Framework Design
- b. Automation Approach
- c. How to scale Automation?
- d. Explanation of Hybrid Framework with a Combination of
  - i. Data-Driven
  - ii. Keyword-Driven
  - iii. Method-Driven
  - iv. Behavior-Driven
- e. Design Principles

- i. SOLID
- ii. DRY
- iii. KISS
- iv. YAGNI
- f. Design Patterns
  - i. Page Object Model Pattern
  - ii. Factory Design Pattern
  - iii. Facade Pattern
  - iv. Singleton Pattern
  - v. Fluent Page Object Model

## 7. BDD using Cucumber

- a. Introduction to Cucumber and BDD
  - i. Overview of BDD and its benefits
  - ii. Introduction to Cucumber and its Architecture
  - iii. Understanding the Gherkin syntax
- b. Writing Feature Files
  - i. Creating feature files for different scenarios
  - ii. Defining scenarios and steps in Gherkin syntax
  - iii. Best practices for writing feature files
- c. Implementing Step Definitions
  - i. Creating step definitions in Java
  - ii. Mapping Gherkin syntax to Java code
  - iii. Implementing parameterization in step definitions
- d. Running Cucumber Tests
  - i. Running Cucumber tests from the command line
  - ii. Generating HTML reports for Cucumber tests
  - iii. Debugging Cucumber tests
- e. Integrating Cucumber with Selenium
  - i. Introduction to Selenium WebDriver
  - ii. Integrating Selenium with Cucumber for UI testing
  - iii. Implementing Page Object Model (POM) with Cucumber and Selenium
- f. Integrating Cucumber with TestNG
  - i. Integrating TestNG with Cucumber for test automation
  - ii. Writing maintainable and scalable Cucumber tests
- g. Advanced Cucumber Topics
  - i. Working with data tables in Cucumber
  - ii. Using hooks in Cucumber
  - iii. Cucumber Reports
  - iv. Integration with Allure reports
  - v. Integration with Extent reports
- h. Final Project
  - i. Students will complete a final project that demonstrates their understanding of Cucumber in BDD. The project will involve writing a set of feature files, implementing step definitions, and running the tests using Cucumber. The project will be graded based on functionality, design, and code quality.

## 8. API Testing with Postman

- a. Introduction to API Testing with Postman
  - i. Overview of API testing and its benefits
  - ii. Introduction to Postman and its Architecture
  - iii. Understanding the Postman environment setup
- b. Setting Up Postman and Creating Requests
  - i. Installing and setting up Postman on Windows/Mac
  - ii. Creating and sending requests with Postman
  - iii. Understanding HTTP request methods, headers, and parameters
- c. Creating and Managing Collections
  - i. Creating collections in Postman
  - ii. Managing requests and responses within collections
  - iii. Using variables and environments in Postman
- d. API Testing with Postman
  - i. Writing API tests in Postman
  - ii. Understanding the Postman testing framework
  - iii. Implementing assertions in API tests
- e. Advanced Postman Features
  - i. Using Postman for exploratory API testing
  - ii. Using Postman for performance testing
  - iii. Understanding Postman's built-in features for load testing
- f. Integrating Postman with Newman
  - i. Overview of the Newman command-line tool
  - ii. Using Newman to run Postman collections
  - iii. Integrating Newman with Github Actions for continuous integration and delivery
- g. Advanced Topics in API Testing
  - i. Understanding and handling authentication in APIs
  - ii. Working with APIs that require authorization tokens
  - iii. Using Postman for API Documentation
- h. Final Project
  - i. Students will complete a final project that demonstrates their understanding of API testing with Postman. The project will involve creating a set of API tests, integrating them with Newman, and running the tests using GitHub Actions. The project will be graded based on functionality, design, and code quality.

## 9. API Test automation using RestAssured

- a. Introduction to API Testing with RestAssured
  - i. What is Rest Assured?
  - ii. Introduction to Rest Assured and its Architecture
  - iii. Understanding Rest Assured Environment Setup
- b. HTTP Request and Response in RestAssured
  - i. Understanding HTTP methods, Headers, Parameters, Request Body & Response Body
  - ii. Understanding API Documentation
  - iii. Understanding Rest-Assured Testing Frameworks
- c. API Testing with RestAssured
  - i. API test to verify GET call using BDD format of Rest Assured
  - ii. API test to verify GET call using Request Specification & Response Specification in Rest Assured
  - iii. Integrating TestNG to run Rest Assured tests
  - iv. API test using Authentication Key to verify GET call
  - v. Automating POST call with the body as Java String
  - vi. Automating POST call with the body as JSON Object
  - vii. Automating PUT & DELETE call
  - viii. Handling Authentications in API using Token.
  - ix. End to End test case to verify Create, Update, DELETE operations of API.
- d. Integrating RestAssured with TestNG & Cucumber
  - i. Integrating Rest Assured with Cucumber
  - ii. Automating GET, POST, PUT, and DELETE calls using Rest Assured with Cucumber
  - iii. Creating Api Utils class to increase reusability and enhance the framework
- e. Final Project
  - i. Students will complete a final project demonstrating their understanding of API test automation using RestAssured. The project will involve creating a set of API tests, integrating them with TestNG, and running the tests using Github Actions. The project will be graded based on functionality, design, and code quality.

## 10. Version Control, Git & Github

- a. What is VCS?
- b. Different types of VCS present
- c. What is Git? & Why choose Git?
- d. What is GitHub?
- e. Difference between Git & GitHub
- f. Difference between Local and Remote Branches
- g. Installation of Git
- h. How to create a GitHub account
- i. GitHub dashboard overview
- j. Creating and adding a public ssh key

- k. How to create a GitHub repository and different modules to look after while creating a repo.(public/private, readme.md, gitignore, protecting the branch)
- l. Git Commands:
  - i. How to clone the repo?
  - ii. How to create a branch & checkout?
  - iii. How to pull the latest code to create a branch?
  - iv. How to commit and push the changes?
  - v. How to delete the branch?
  - vi. How to reverse the commit?
  - vii. Other basic git commands
- m. What is a pull request?
- n. How to create a pull request?
- o. What are merge conflicts?
- p. How to resolve the merge conflicts?

## 11. DevOps & CI/CD

- a. What is DevOps?
- b. Why is it necessary for testers?
- c. What is CI/CD?
- d. Benefits of CI/CD
- e. Different tools used for CI/CD
- f. The CI/CD process flow
- g. Introduction of GitHub Actions for CI/CD
- h. What is workflow?
- i. YAML language introduction and structure
- j. Overview of GitHub marketplace
- k. How to create a workflow file?
- l. What are the different components of CI/CD workflow?
- m. Difference between self-hosted and GitHub cloud-hosted runner
- n. Create a complete CI/CD workflow file for your repository

## 12. Database & SQL

- a. Introduction to Databases
  - i. Introduction to Databases and their importance
  - ii. Understanding different types of databases
  - iii. Overview of database management systems
- b. SQL Basics
  - i. Introduction to SQL and its importance
  - ii. Understanding SQL syntax and statements
  - iii. Creating tables and basic operations on tables
- c. SQL Joins and Subqueries

- i. Understanding SQL joins and their types
        1. Cartesian Join
        2. Inner Join
        3. Outer Join
        4. Self-Join
      - ii. Subqueries and their practical use
    - d. Overview of SQL functions and their types
      - i. Single Row Functions
      - ii. Multi Row Functions
        1. Max()
        2. Min()
        3. Sum()
        4. Avg()
        5. Count()
    - e. Data Query Language (DQL)
      - i. Select
      - ii. From
      - iii. Where
      - iv. Group By
      - v. Having
      - vi. Order By
      - vii. Advanced SQL queries and their practical use
    - f. Database Design and Normalization
      - i. Understanding database design principles
      - ii. Normalization techniques and their importance
      - iii. Designing efficient and scalable databases
    - g. Final Project
      - i. Students will complete a final project that demonstrates their understanding of database design and SQL queries. The project will involve designing a database, creating tables and relations, and implementing SQL queries to query and manipulate data in the database. The project will be graded based on functionality, design, and code quality.

### 13. ALM using Jira

- a. Introduction to ALM
  - i. Overview of Application Lifecycle Management
  - ii. Understanding Jira platform and integrations
- b. Creating and Managing Projects
  - i. Creating a project in Jira
  - ii. Understanding project components and configurations
  - iii. Configuring project permissions and roles
- c. Creating and Tracking Issues
  - i. Creating and managing issues in Jira
  - ii. Understanding issue types and their uses
  - iii. Creating custom fields and workflows
- d. Performing Version Control with Jira

- i. Introduction to version control
  - ii. Creating and managing versions in Jira
  - iii. Using Jira for release management
- e. Using Agile Methodologies in Jira
  - i. Using Jira for agile project management
  - ii. Creating agile boards and using them for project tracking
- f. Tools and Techniques for Efficient Project Management
  - i. Using Jira filters and reports for project management
  - ii. Understanding and using Jira dashboards
  - iii. Best practices for project management with Jira
- g. Integrating Jira with Other Tools
  - i. Integrating Jira with other development and testing tools
  - ii. Understanding and using Jira plugins
  - iii. Best practices for integrating Jira with other tools

#### 14. Test Management Tool - Xray

- a. Introduction to Test Management
  - i. Overview of Test Management
  - ii. Understanding Xray platform and licensing
- b. Creating and Managing Test Cases
  - i. Creating and managing test cases in Xray
  - ii. Understanding test case templates
  - iii. Creating custom fields and workflows
- c. Creating and Managing Test Plans
  - i. Creating and managing test plans in X Ray
  - ii. Understanding test plan templates
  - iii. Creating custom fields and workflows
- d. Executing Tests in X Ray
  - i. Executing tests in X Ray
  - ii. Understanding test execution status and results
  - iii. Defining test preconditions and postconditions
- e. Generating Reports in X Ray
  - i. Using X Ray reporting features
  - ii. Understanding X Ray report templates
  - iii. Creating custom reports
- f. Integrating X Ray with Other Testing Tools
  - i. Integrating Xray with Jira and other development tools
  - ii. Understanding and using Xray REST API
  - iii. Best practices for integrating X Ray with other tools
- g. Advanced Topics in X Ray
  - i. Test Automation with X Ray
  - ii. Customizing X Ray workflows and screens