



# Master in Software Testing Curriculum

## STRUCTURE



## Master in Software Testing Training Curriculum

*“Transform your Career with Software Quality Testing Training at Croma Campus”*

### **About Croma Campus:**

Croma Campus Training & Development Private Limited is an education platform since 2010 providing rigorous industry-relevant programs designed and delivered in collaboration with world-class faculty and industry.

- Hands-On Live Projects
- Simulation Test Papers
- Industry Cases Studies
- 61,640+ Satisfied Learners
- 140+ Training Courses
- 100% Certification Passing Rate
- Live Instructor Classroom / Online Training
- 100% Placement Assistance

### **Course Objectives:**

- Learn software testing fundamentals, manual testing, database testing, and basic mobile testing concepts.
- Prepare yourself for ISTQB certification exam and clear it in first attempt.
- Project yourself as a skilled software tester and a knowledgeable quality expert.
- Understand the key issues in testing software applications.
- Learn how to design tests that adequately cover requirements and business events.
- Advance your career by reinforcing your testing expertise.
- Learn from an industry recognized expert in software testing and quality
- Start applying for jobs and get hired by top MNCs worldwide.

### **Course Description:**

Our Software testing training program is suitable for testers, test analysts, test engineers, test consultants, test managers, user acceptance testers and software developers. This Foundation Level qualification is also appropriate for anyone who wants a basic understanding of software testing, such as project managers, quality managers, software development managers, business analysts, IT directors and management consultants.

### **Croma Campus Training Program Deliverables:**

- **Session Recordings** - Original Class Room Voice & Video Recording
- **Training Material** - Soft Copy Handbooks
- **Assignments** | Multiple Hands-on Exercises
- **Test Papers** - We provide **Practice Test** as part of our course to help you prepare for the actual certification exam.
- **Live Case Studies**
- **Live Projects** - Hands-on exercises and Project work. You will work on real time industry-oriented projects and assignments for each module to practice.
- **Key focus on Hands-on exercises and Project work.** You will work on real time industry-oriented projects.

- Faculty with more than **10+ Years of Experience** in the Industry.
- **Technical Resume Designing & Job Assistance:** With more than 100+ Clients across the Globe and we help learners to get a good job in their respective field. We also help learners with resume preparation.
- **Interview Q&A**
- **About Croma Campus Training Certificate:** Croma Campus will provide you with an industry-recognized (Certified by **ISO 9001:2015 & E-Cell IIT Jodhpur**) course completion certificate which has lifelong validity.
- **How I Unlock my Croma Campus Certificate:** Attend Complete Batch & Submit at least One Completed Project.

### Program Modules:

- **Manual Software Testing**
- **ISTQB Certification Training**
- **Database Testing Training**
- **Agile and Scrum Training**
- **Selenium Certification Training**
- **Appium Testing Training**
- **JMeter Certification Training**
- **Postman API Testing Training**
- **AWS Cloud Certification Training**

## *Master in Software Testing Course Content*

### **Course 1. Manual Software Testing**

#### **Module 1: Software Testing Fundamentals**

- Introduction
  - What is Software Testing?
  - Why do we need Testing?
  - How to perform the Testing?
  - Benefits of Testing
- Software Testing Terminologies
  - Mistakes, Faults, Bugs & Failures
  - Priority & Severity
  - Debugging & Root Cause Analysis
  - Verification & Validation
  - Test Bed/Environment
  - Mock-ups
  - Test Scenario
- Quality Assurance & Quality Control
  - What is Quality Assurance?
  - What is Quality Control?
  - Quality Assurance vs Quality Control
- Software Testing as a Career
  - Best testing practices used in Software Industry
  - Software Tester Roles
  - Job Responsibilities of a Software Tester
  - Why choose Software Testing as a career?
  - How training can help you to become a Software Tester?

#### **Module 2: SDLC vs STLC**

- Requirement and Analysis
  - Requirement Gathering
  - SRS Creation
  - Feasibility Study
  - Requirement Validation
  - SRS Approval
  - Project Kick-off
  - Application Walkthrough
  - Requirement Analysis
- Design Concepts in Software Testing
  - System Design
  - Architecture Design
  - Module Design
- Coding/Development
- Testing
- Implementation/Live

- Software Testing Life Cycle (STLC)
  - Requirements analysis
  - Test planning
  - Test development
  - Test execution
  - Test reporting
  - Test result analysis or Defect Analysis
  - Retesting
  - Regression testing
  - Test Closure
- SDLC Vs STLC
  - What is The Difference?
  - Which is better? What to choose?

### **Module 3: Software Testing Models and Methodologies**

- Software Testing Models
  - Waterfall Model
  - Iterative Model
  - Prototyping Model
  - Progressive Model
  - Spiral Model
  - V-Model
  - RAD (Rapid Application Development)
- Agile Methodology?
  - What is Agile?
  - Why choose Agile Methodologies?
- Scrum Role in Agile
  - Product Owner
  - Scrum Master
  - Scrum Team
- Work Products
  - Product Backlog
  - Spring Backlog
  - Burndown Chart
- Ceremonies
  - Sprint Planning Meeting
  - Daily Scrum Call
  - The Sprint
  - Sprint Review
  - Sprint Retrospective
  - Scrum of Scrums
- Testing Methodologies
  - Black Box Testing
  - Boundary Value Analysis
  - White Box testing
  - Path Testing

- Mutation Testing

## **Module 4: Testing Levels and Testing Types in Software Engineering**

- Testing Levels
- Testing Types
- How to choose them?
- Signification of Different Testing Types
- Unit Testing
- Integration Testing
- Re Testing
- Regression Testing
- Usability Testing
- Performance Testing
- Software Stress Testing
- Security Testing
- Smoke Testing
- Compatibility Testing
- System Testing
- UAT (User Acceptance Testing)
- Alpha & Beta Testing

## **Module 5: Test Cases and Test Plans in Testing**

- Test Cases
- Review Process
  - Peer Review
  - Walkthrough
  - Inspection
- Bug/Defect Management
  - Bug/Defect Lifecycle
  - Bug Tracking through Tool (Jira/Bugzilla)
- Test Plan
  - Introduction
  - Test Objective
  - Test Environment
- Scope of Testing
  - Modules to be Tested
  - Modules not to be Tested
  - Types of Testing
  - Levels of Testing
- Testing Schedule
  - Entry Criteria
  - Testing Flow
  - Test Deliverables
  - Defect Tracking
  - Task Management

- Suspension/Resumption Criteria
- Risk & Mitigation Plan
- Completion/Exit Criteria
- Test Design Checklist
- Test Execution Checklist
- Cross Browser Testing

## **Module 6: Project Life Cycle Management/Live Project**

- Introduction to Project Life Cycle
  - Identify Testing Artifacts
  - Test Strategy
  - Test plan
  - Software Test Life Cycle (STLC)
  - Prepare Test Case
  - Optimizing Test Cases (Testing Techniques)
  - Bug Reporting and Management
  - Application Sign Off
- Live Project
  - Requirement Analysis
  - Prepare Query Log
  - Writing Test Cases
  - Review of Test Cases
  - Execution of Test Cases
  - Bug Finding & Reporting
- ISO and CMMI Certifications, its significance

## **Module 7: ISTQB Certification**

- Software Testing Fundamentals
  - Why is Testing Necessary
  - What is Testing?
  - Seven Testing Principles
  - The Psychology of Testing
  - Code of Ethics
- Testing Throughout the Software Life Cycle
  - Software Development Models
  - Test Levels
  - Test Types
  - Maintenance Testing
- Static Techniques
  - Static Techniques and the Test Process
  - Review Process
  - Static Analysis by Tools
- Test Design Techniques
  - The Test Development Process
  - Categories of Test Design Techniques
  - Specification-based or Black-box Techniques

- Structure-based or White-box Techniques
- Experience-based Techniques
- Choosing Test Techniques
- Test Management
  - Test Organization
  - Test Planning and Estimation
  - Test Progress Monitoring and Control
  - Configuration Management
  - Risk and Testing
  - Incident Management
- Tool Support for Testing
  - Types of Test Tools
  - Effective Use of Tools: Potential Benefits and Risks
  - How to Introduce a tool into an Organization?

## **Module 8: Mobile Testing Basics**

- Mobile Testing Introduction
  - Overview of Mobility Testing
  - Mobile Application Development Process
  - Introduction to various Mobile platforms
  - Introduction to Various Apps, Mobile App, Web based App, Native App, Hybrid App.
- Mobile Testing Strategy
  - Device Emulator
  - Mobile Cloud Computing (Remote Real Devices)
  - Real Time Devices
  - Automation Tools
  - Emulators and its usage
  - Installation and un- installation of Android Emulator
- Mobile Testing Approach
  - Mobility Application – Test Approach/Phases
  - Test Requirements
  - Test Planning
  - Test Environment setup
  - Test Design
  - Test Execution and Defect tracking
  - Test Reports and Acceptance
- Mobile Testing Scope
  - Unit testing
  - Integration testing
  - Functional testing
  - Device conformance testing
  - System testing
  - Regression testing
  - Mobile browser compatibility testing
  - Performance testing



## Course 2. ISTQB Certification Training

### Course Description:

The main objective of ISTQB training course helps to learn all the basic testing concepts and methodologies. You can learn to employ a common communication language while working and how to communicate with other testers.

### Module 1: Fundamentals of Testing

- Fundamentals of Testing
  - What is Testing
    - Typical Objectives of Testing
    - Testing and Debugging
  - Why is Testing Necessary
    - Testing's Contributions to Success
    - Quality Assurance and Testing
    - Errors, Defects, and Failures
    - Defects, Root Causes and Effects
  - Seven Testing Principles
  - Test Process
    - Test Process in Context
    - Test Activities and Tasks
    - Test Work Products
    - Traceability between the Test Basis and Test Work Products
  - The Psychology of Testing
    - Human Psychology and Testing
    - Tester's and Developer's Mindsets

### Module 2: Testing Throughout the Software Development Lifecycle

- Software Development Lifecycle Models
  - Software Development and Software Testing
  - Software Development Lifecycle Models in Context
- Test Levels
  - Component Testing
  - Integration Testing
  - System Testing
  - Acceptance Testing
- Test Types
  - Functional Testing
  - Non-functional Testing
  - White-box Testing
  - Change-related Testing
  - Test Types and Test Levels
- Maintenance Testing
  - Triggers for Maintenance
  - Impact Analysis for Maintenance

## Module 3: Static Testing

- Static Testing Basics
  - Work Products that Can Be Examined by Static Testing
  - Benefits of Static Testing
  - Differences between Static and Dynamic Testing
- Review Process
  - Work Product Review Process
  - Roles and responsibilities in a formal review
  - Review Types
  - Applying Review Techniques
  - Success Factors for Reviews

## Module 4: Test Techniques

- Categories of Test Techniq
  - Categories of Test Techniques and Their Characteristics
- Black-box Test Techniques
  - Equivalence Partitioning
  - Boundary Value Analysis
  - Decision Table Testing
  - State Transition Testing
  - Use Case Testing
- White-box Test Techniques
  - Statement Testing and Coverage
  - Decision Testing and Coverage
  - The Value of Statement and Decision Testing
- Experience-based Test Techniques
  - Error Guessing
  - Exploratory Testing
  - Checklist-based Testing

## Module 5: Test Management

- Test Management
  - Test Organization
    - Independent Testing
    - Tasks of a Test Manager and Tester
  - Test Planning and Estimation
    - Purpose and Content of a Test Plan
    - Test Strategy and Test Approach
    - Entry Criteria and Exit Criteria (Definition of Ready and Definition of Done)
    - Test Execution Schedule
    - Factors Influencing the Test Effort
    - Test Estimation Techniques
  - Test Monitoring and Control
    - Metrics Used in Testing
    - Purposes, Contents, and Audiences for Test Reports

- Configuration Management
- Risks and Testing
  - Definition of Risk
  - Product and Project Risks
  - Risk-based Testing and Product Quality
- Defect Management

## **Module 6: Tool Support for Testing**

- Test Tool Considerations
  - Test Tool Classification
  - Benefits and Risks of Test Automation
  - Special Considerations for Test Execution and Test Management Tools
- Effective Use of Tools
  - Main Principles for Tool Selection
  - Pilot Projects for Introducing a Tool into an Organization
  - Success Factors for Tools

## **Course 3. Database Certification Testing Training**

- What is database testing?
- Database testing through SQL
- What is SQL?
- Basic SQL Queries
- Create SQL Queries
- Writing SQL Queries
- SQL Where Clause
- AND/OR clauses
- Order By/Update/Delete database
- Primary Key vs Foreign Key
- Joining Tables
- LEFT/RIGHT JOIN
- Joining more than two tables- complex queries
- Using Aggregation to determine data health
- Foreign Key Problems
- Using Sub-queries
- Generating Test Data
- Views and Indices

## **Course 4. Agile and Scrum Training**

### **Course Description:**

This Agile Scrum Master Training introduces the best practices of Agile and the nuances of the Scrum methodology. Master the Agile Scrum project management approach and enhance your

ability to develop and deliver quality products to customers. Use this popular methodology to maximize business value while mitigating potential risks.

This Agile Scrum Master training enables you to deliver products in functional stages, improve collaborations, and create a flexible process where changes are less impactful on project quality, cost, and timing.

The Agile Scrum Master course is best suited for Team leads, Project managers, Members of Scrum teams such as developers, Scrum Masters, and product owners, Managers of Scrum teams, Teams transitioning to Scrum, and Professionals intending to pursue the Agile Scrum Master training.

All the best for a new beginning and successful career ahead!

## **Course Content:**

### **Module 1: Introduction**

- Introduction to Agile/Scrum
- Why Agile?
- What makes Scrum adoptions successful
- How do you know/measure it was successful?

### **Module 2: Other Agile Frameworks**

- Different development methodologies
- Waterfall, XP, Crystal, DSDM, Lean
- Linking Agile to DevOps and IT Service Management

### **Module 3: Scrum Events and Artifacts**

- Three Pillars of Scrum
- Scrum Values
- Scrum Life Cycle
- Sprint
- Three Scrum Backlogs
- Product Backlog
- Four Scrum Ceremonies
- Definition of Done

### **Module 4: Scrum Roles**

- Scrum master and product owner roles
- Attributes of good scrum masters
- Product owners
- Failure mode
- Team manager or developer role

### **Module 5: Agile Estimating, Planning, Monitoring, and Control**

- User Stories and Epics
- Characteristics of Good User Stories

- Story Card
- Splitting Stories
- Determining a Value or Return on Investment (ROI)
- Prioritization Models
- Velocity
- Planning Onion
- Roadmap/Release Planning
- Output of Release Planning
- Estimation
- Cone of Uncertainty
- Comparing Ideal Time with Story Points
- Planning Poker
- Affinity Estimation
- Tracking Releases and Sprints
- Physical Progress Chart
- Communicating Progress
- Staying in Control

## **Module 6: Agile on Complex Projects**

- Agile at scale
- Scaled agile foundation introduction
- Where to apply agile?
- Tools for agile project management

## **Module 7: Adopting Agile**

- Adopting Agile
- Transition plan for Agile transformation
- Towards self-organization
- Physical and cultural changes needed

## **Course 5: Selenium Certification Training**

### **Module 1: Java Course Contents**

- Introduction to java
  - What is java?
  - Programming language hierarchy
  - Java files
  - Camel casing
  - Identifiers
  - Java Components naming standards
  - Java source file declaration rules
  - First Java program
- Classes and Objects
  - Introduction to Classes and objects
  - Understanding Java Heap

- Creating first Class and Object
- Variables and Operators
  - Introduction to Variables and their data types
  - Primitive and Non-primitive variables
  - Variable Casting
  - Object references
  - Java Operators
- Java Methods and their communication
  - Introduction to methods
  - Method arguments and return types
  - Pass by Value
  - Encapsulation
  - Getters and Setters
- Loops and Arrays
  - If-else statement
  - While loop
  - Do-while loop
  - For loop
  - Enhanced for loop
  - Arrays – 1D and 2D
  - Reference in an array
- Understanding Java-API
  - Understanding API using ArrayList
  - ArrayList fundamentals
  - Using Java Library
  - Using Packages
  - Using HTML- API docs
- Inheritance and Polymorphism
  - Understanding inheritance and inheritance tree
  - Methods overriding and the rules
  - IS-A and HAS-A relationship
  - Super class Vs Subclass
  - Method Overloading
  - Access Modifiers
- Abstract Classes and Interfaces
  - Abstract classes and methods
  - Mother of all classes – “Object class”
  - Polymorphic reference
  - Object reference casting
  - Deadly Diamond of Death
  - Interfaces and it’s implementation
- Garbage Collection and Constructors
  - Concept of Stack and Heap
  - Methods and classes on Stack and Heap
  - Constructors
  - Constructor Overloading
  - Constructor chaining and this() keyword
  - Garbage collection eligibility

- Statics and data formatting
  - Static methods, variables and constants
  - Math class and methods
  - Wrapper classes
  - Auto boxing
  - Data formatting and static imports
- Exception handling
  - Risky java codes
  - Introduction to Java Exceptions
  - Catching exceptions using try/catch block
  - The finally block
  - Catching multiple exceptions
  - Handle or Declare law of exceptions
- Serialization and file I/O
  - Saving objects state
  - Writing serialized object
  - I/O streams
  - Serializable interface
  - Deserialization
  - Java IO file
  - Reading/Writing a text file
- Collections
  - Introduction to Java collections
  - Concept of generics
  - Lists
  - Sets
  - Map
  - Hashmap
- JDBC
  - JDBC concepts and terminology
  - JDBC package
  - Creating connection to a DataBase
  - Accessing and manipulating tables using JDBC
  - Sql statements

## **Module 2: Selenium WebDriver3.0**

- Introduction to Selenium WebDriver
  - What is WebDriver
  - Exploring Webdriver java docs
  - Downloading Webdriver jar files
  - Open Google home page through WebDriver
  - Exploring more features of WebDriver
  - WebDriver Features & Examples
  - Handling HTML Elements such as: Text box, Hyperlinks, Submit buttons,
  - Radio buttons, Check boxes, Dropdown etc
  - Finding Elements & text on a Webpage
  - Printing all links and other values from a webpage and HTML Elements

- Using By class to find elements using different methods
- Implement global wait
- Running test in multiple browsers
- Firefox profile
- Running test in multiple profiles
- Advance WebDriver Features
  - Simulating keypress events such as buttons such as Enter, page up, pagedown, backspace
  - Handling java script messages
  - Working with online captchas
  - Running the test with HTMLUnit driver
  - Online Webtables
  - Downloading and configuring chrome driver
  - How to find Xpaths through Chrome developer tool
  - Generating your own Xpaths
  - What if you don't have firebug or firepath
  - Handling multiple pop ups/tabs in Chrome
  - Handling Mouse over Menus and Iframes
- Ajax & Listeners
  - How to handle Ajax based applications
  - Browser navigate methods
  - Listeners
  - Mouse movements
  - Explore action class mouseover method
  - Working with Javascriptexecutor
    - Drag and Drop
    - Resizable
    - Handling Sliders
- Flash Testing with Selenium WebDriver
  - What is flash/flex testing
  - Downloading and installing flash jar file
  - Automating flash player
  - Using different call methods of YouTube flash player

### **Module 3: TestNG Framework**

- Integration and Execution of Test Suite using TestNG Framework
  - What is TestNg
  - Installing TestNg in Eclipse
  - TestNg Annotations
  - Understanding usage of annotations
  - Running a Test in TestNg
  - Batch Running of tests in TestNg
  - Skipping Tests
  - Parameterizing Tests – DataProvider
  - Assertions/Reporting Errors
  - TestNg Reports
  - Advantages over Junit



## Module 4: ReportNG Framework

- ReportNG is a simple plug-in for the TestNG unit-testing framework to generate HTML reports as a replacement for the default TestNG HTML reports.
- Configuring ReportNG with TestNG for HTML Reports

## Module 5: Selenium Grid - Cross Platform & Parallel Execution

- Cross Browser and Cross Platform Testing using Web Driver
  - What is Selenium-Grid?
  - How Selenium-Grid Works–With a Hub and Nodes
  - Configuring Selenium-Grid
  - Default Configuration
  - JSON Configuration File
  - Configuring Via the Command-Line Options
  - Node Configuration
  - Timing Parameters

## Module 6: Robot Class

This class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations. Using the class to generate input events differs from posting events to the AWT event queue or AWT components in that the events are generated in the platform's native input queue. For example, Robot.mouseMove will actually move the mouse cursor instead of just generating mouse move events.

## Module 7: Integration of Selenium with Jenkins

In Selenium Training, Croma Campus will include the latest implementation of automation concepts like: Integration of Jenkins and Selenium Automation testing. Jenkins is a powerful and highly configurable continuous integration tool (CI) tool that is commonly used on IT projects to manage builds and releases. Selenium is a powerful suite of tools that automates web browser testing. Our training will guide you through the steps required to get you started with Jenkins and Selenium from an automation tester's perspective. During our training we will cover the following aspects:

- Installation and setup of Jenkins & Selenium project on your laptop.
- Jenkins Overview.
- Jenkins Plugins.
- Jenkins Configuration for a Java-based Project.
- Integration of Selenium and Jenkins CI.
- Reporting and Validation in Jenkins.
- Continuous integration and continuous testing
- Test Execution.
- SVN, Junit, Source Control, Ant, Maven Java integration.

## Module 8: Data Driven & Hybrid Test Automation Framework

Covers end to end steps followed while creating the framework:

- Test Data files
- TestCore class which loads Xls file, run selenium server through code and do other initialization
- Object.Properties file to store Xpaths
- Configuration Files
- Skipping Test Cases
- Screenshot Capturing
- Emailing Test Results
- Generating Reports
- Generating Application and Selenium logs
- Running framework automation through Maven

## **Module 9: Database Testing - Java DataBase Connectivity JDBC**

- Installing MySQL Database
- Java.Sql Package |JDBC Drivers
- Connection Interface
- Statement and Prepared Statement Interfaces
- ResultSet Interface
- Firing Select, Insert, Update and Delete queries with database using Java JDBC
- Looping the ResultSet
- Using Annotations of Junit/TestNg to establish connection with database when Using Selenium

## **Course 6: Appium Testing Training**

### **Module 1: Introduction to Appium**

- Introduction to Mobile Automation
- Various tools for mobile application?
- What is Appium?
- What all can be tested with Appium?
- WebDriver Wire protocol
- Hybrid Apps, Native Apps and WebApps
- Concept of UIAutomator
- When to use Selendroid and when to use Appium
- IOS-driver
- Drawbacks and Limitations
- What is selendroid
- Difference between Appium and Selendroid
- How things came into being
- What should I learn?
- Appium Internal Architecture

### **Module 2: Download and Configure Appium/Maven Dependency**

- Configuring Appium on real device
- Why work on real device?
- Download and Configure Android on local machine
- Understanding difference between Api level and Android version
- API levels and Android versions supported by Appium supported by android
- Getting correct API level for Android
- Getting correct Android version in Phone – Provider dependant
- Enabling developer mode
- Debug level settings in Phone
- Connecting phone and making sure it's being detected in DDMS(Pdanet software)
- Install Node JS
- Install Microsoft Dot Net framework
- Download Appium
- Configuring Various versioning/API level requirements for Appium
- Starting Appium Server from Console
- Starting Appium Server from command prompt
- Various command line arguments for Appium
- Running your first Appium Program
- Configuring Appium on Simulator
- Installing Simulator
- Running first Script on Simulator
- Maven dependencies
- Various Maven/POM dependencies for Appium
- Downloading/Configuring Appium, Selenium from Maven/POM dependencies

### **Module 3: Launching Hybrid/Native Apps with Appium**

- Desired Capabilities for Appium and Android
- Concept of Apk file, AppCompatActivityName and AppPackage
- How to get Apk file, AppCompatActivityName and AppPackage for any application
- Is Apk file necessary to test App – Not in all cases
- Javadocs for Appium
- AppiumDriver and RemoteWebDriver classes
- Launching System Hybrid Apps
- Appium reinstalls App
- Preventing Appium from deleting and reinstalling App
- Launching other user installed Hybrid Apps
- Launching native Apps like calculator, Contacts, Settings
- Always quit Appium – Mandatory

### **Module 4: Identifying Elements-1**

- Concept of UI Automator
- Understanding Android Layouts
- Tag name not supported

- android.widget.\* – everything starts from here
- Find Element and FindElementByAndroidUIAutomator
- Using UISelector Class in UIAutomator API
- Finding by Classname
- Detecting presence of Elements
- Building Xpaths
- Challenges faced in finding Elements and Workarounds
- Using Android UI Automator's UISelector() and its functions
- Hiding Keyboard
- Waiting – Implicit and explicit waits
- Using UI Automator in Compressed layout

## **Module 5: Identifying Elements-2**

- Finding multiple Elements
- Finding Elements inside Elements
- ComplexFind and Mobile Element
- Finding Elements with Multiple criteria using Complex Find
- Selector Key codes
- Finding elements by text using AndroidUIAutomator commands
- Using Assertions
- Handling Date Elements/Objects
- Selecting element in Spinner
- Finding selected Elements
- Getting Value from Text View
- Finding selected Elements & Clear value of a Text View

## **Module 6: XML Viewer**

- Getting and storing the Android App XML view
- When to use android xml view
- Launching XML View from WebDriver
- Finding Elements in XML view
- Finding Scroll List elements in XML View
- you can do anything with XML View
- Limitations of XML view
- Touch Actions
- Touch Events
- Android key events like home, backspace, delete etc.
- Wait Actions
- Dragging and dropping elements
- Multitouch Actions
- Simulating multiple gestures
- Longpress and delete elements
- Handling Toggle buttons

- Pinching, tapping, holding

## **Module 7: Web Application Testing**

- Launching browser with Appium
- Making sure you have the correct desired capabilities
- When launch fails
- Firing various commands on browser
- Will UIAutomator work here?
- How do I identify objects in mobile browser?
- Will AppiumDriver commands work on browser?
- WFinding and setting user agent

## **Module 8: Appium First Program**

- Android App download for practice
- What are desired capabilities? How to talk to appium server
- Invoking Android Driver- Creating base program
- Execution of Appium first program on Mobile Native APP
- Appium desired capabilities code download
- UIAutomator tool usage-Inspection of elements

## **Module 9: Native Apps Automation**

- Automating app UI Using Xpaths and text attributes
- Appium API's for UI interaction with id's and classNames
- Handling Mutiple similar objects of Apps with indexes
- AndroidUIAutomator usage in identifying objects of Apps
- TouchActions in Appium (Gestures Automation -1)
- Tapping and longpress events (Gestures Automation -2)
- Strategy for automating Swipping event(Gestures Automation -3)

## **Module 10: Advanced Gestures Automation with Key Events Handling**

- Events Handling
- Demo on swapping the clock- Practice exercises
- How to automate scrolling? Gestures Automation -5
- Drag and drop on Native Apps with Appium (Gesture Automation -6)
- Automating Android Key Events with Appium
- Miscellaneous key Events handling with Appium
- Invoking App with package Activity
- Example on package name and Activity Invoking apps
- Automation on real devices

## **Module 11: Web Apps Automation**

- Mobile Browser Configuration setup for Appium Tests
- Code for Mobile Browser capabilities

- Automating the Chrome Mobile Browser

## **Module 12: Practical Problems and solutions with Mobile Browsers**

- Identifying frames from Html view
- Element Hidden or Invisible-How to validate
- Writing Generic functions to Identify Frames/Windows
- Example Demonstrating Multiple Frames
- Handling Auto suggestive dropdown options
- Dynamic data loading- Exception Techniques

## **Module 13: Hybrid Apps Automation**

- Hybrid Apps features and ways to test them
- Views switching Mechanism
- Example on Hybrid App handling

## **Module 14: Appium Framemwork-Learn TestNG**

## **Module 15: Integrating TestNG with Maven**

## **Module 16: Pageobject Pattern & Page Factory**

## **Module 17: Data driving from Excel for feeding data into Appium Test Cases**

## **Module 18: Logging Framework - log4j**

## **Module 19: Database connection to Selenium/Appium Test Cases**

## **Course 7: JMeter Certification Training**

### **Course Description:**

This course will help you to gain in-depth knowledge of various functionalities of JMeter to evaluate the performance of an application. You will be able to check the performance of an application under different workload conditions and understand different methodologies related to the security of an application. This course designed to help you become a certified practitioner through intensified training in the best practices for checking the efficiency of software.

The course provides you insights into software behavior during workload. In this course, you will learn how to check the response time and latency of software and test if a software package is efficient for scaling. The course will help you check the strength and analyze the overall performance of an application under different load types.

### **Course Content:**

## **Module 1: Performance/Load Testing Fundamentals**

- About Testing
  - Introduction to Non-Functional testing

- Need for Non-Functional testing
- Types of Non-Functional testing
- Introduction to Performance Testing
- Performance Testing components
- Performance testing and tuning
- Baselines
- The incident
- The aftermath
- Performance Testing Using JMeter
  - JMeter: The preferred tool for Performance Testing
  - Introduction to Apache JMeter
  - Elements of JMeter
  - Performance testing using JMeter
  - Assertions, controllers, and processors in JMeter
  - JMeter best practices
  - JMeter to the rescue
  - Up and running with JMeter
  - Installation
  - Installing the Java JDK
  - Setting JAVA\_HOME
  - Running JMeter
  - Tracking errors during test execution
  - Configuring JMeter
- Load and stress testing
  - What is Load testing?
  - Need for Load testing
  - Objectives of Load testing
  - How to perform Load testing using JMeter
  - What is Stress testing?
  - Need for Stress testing
  - Purpose of Stress testing
  - How to perform Stress testing using JMeter?

## **Module 2: Recording Your First Test**

- Configuring the JMeter HTTP proxy server
- Setting up your browser to use the proxy server
- Using a browser extension
- Changing the system settings
- Running your first recorded scenario
- Anatomy of a JMeter test
- Test Plan
- Thread Groups
- Controllers
- Samplers
- Logic controllers
- Test fragments

- Listeners
- Timers
- Assertions
- Configuration elements
- Pre-processor and post-processor elements
- Summary

### **Module 3: Submitting Forms**

- Capturing simple forms
- Handling checkboxes
- Handling radio buttons
- Handling file uploads
- Handling file download
- Handling the XML response
- Summary

### **Module 4: Managing Sessions**

- Managing sessions with cookies
- Managing sessions with URL rewriting
- Summary

### **Module 5: Resource Monitoring**

- Basic server monitoring
- Setting up Apache Tomcat Server
- Configuring Tomcat users
- Setting up a monitor controller in JMeter
- Monitoring the server with a JMeter plugin
- Installing the plugins
- Adding monitor listeners to the test plan
- Summary

### **Module 6: Distributed Testing**

- Remote testing with JMeter
- Configuring JMeter slave nodes
- Configuring one slave per machine
- Configuring the master node to be tested against one slave per machine
- Configuring multiple slave nodes on a single box
- Configuring the master node to be tested against multiple slave nodes on a single box
- Executing the test plan
- Viewing the results from the virtual machines
- Summary

### **Module 7: Timers and Functions**



- JMeter properties and variables
- JMeter functions
- The Regular Expression tester
- The Debug sampler
- Using timers in your test plan
- The Constant timer
- The Gaussian random timer
- The Uniform random timer
- The Constant throughput timer
- The Synchronizing timer
- The Poisson random timer
- The JDBC Request sampler
- Configuring a JDBC Connection Configuration component
- Adding a JDBC Request sampler
- Summary

## Module 8: Helpful Tips

- JMeter integration with selenium
- Handling FTP request
- Installing CA certificate in the browser
- Testing REST web services
- Parameterization using CSV File
- Parameterization using Test Plan
- Summary

## Course 8: Postman API Testing Training

### Course Description:

RESTful APIs (or simply REST API) are everywhere nowadays but at the same time they are getting more complex to get started with: different HTTP methods, headers, cookies, dealing with file uploads or authentication with API keys, tokens, OAuth and so much more.

But before you start investing time writing code to retrieve the data the API is offering, why not test the request first to make sure everything is working as expected?

This is where the Postman App comes in! Postman allows you very quickly create a request with the required HTTP method and parameters, submit the request and easily inspect the results.

Postman can help if you are developing APIs as well! I have created this course for testing engineers and well as for software developers. Postman can help you during the development of your API as well as after the API is completed, by running tests that make sure your API is still working as intended.

In this course, you will explore the features of Postman and continue by writing API tests with the intention of integrating them in a CI server where the tests will run on a current basis.

## **Course Content:**

### **Module 1: Introduction**

- Overview
- First steps in Postman
- The postman Landscape

### **Module 2: Creating with API Requests**

- How to create a request in Postman?
- How to import a request from your browser in Postman?
- How to inspect HTTP responses in Postman?
- Handling cookies in Postman
- Troubleshooting Postman issues
- Saving requests to a collection in Postman

### **Module 3: Writing Tests and Scripts**

- Introduction
- Your first API test in Postman
- Testing an API
- Path parameters vs query parameters
- Refactoring tests

### **Module 4: Writing Tests and Scripts using Variables**

- Variables in Postman
- Global variables
- Environments
- Session variables (new in Postman 6.2!)
- Bulk editing removed (new in Postman 6.2!)
- Pre-request scripts in Postman
- Understanding different variable scopes / types
- How to setup different URLs using environments in Postman?
- Debugging tests

### **Module 5: Building and Testing an API Workflow using Github and Twilio**

- Introduction to Github API
- Introduction to 2FA
- Setting up 2FA on Github
- Setting up 2FA in Postman
- Automating two-factor authentication (2FA) workflow - overview
- Introducing Twilio
- Get a new Twilio phone number & 2FA setup
- Setting up 2FA on Github with Twilio
- Troubleshooting why Twilio did not receive the SMS

- Using the Twillio API in Postman
- Conclusion

## **Module 6: Advanced Assertions**

- Section overview
- Postman assertion basics
- Assertions
- Chai Assertion Library
- Assertions on arrays
- Assertions on nested objects

## **Module 7: Automatically Running Tests**

- Overview
- Postman collection runner
- Postman Monitors
- Automating with Newman (Overview)
- Installing Newman
- Running a collection with Newman
- Newman v3 to v4 Migration Guide
- 3 ways to access your Postman collection from Newman
- Specifying environments with Newman
- Prerequisites for running Newman on Jenkins
- Running a collection with Jenkins and generating a report
- Creating an HTML report in Jenkins
- Generating better HTML reports
- Customizing HTML reports
- Running a collection from a Git repository with Jenkins

## **Module 8: Running Newman with other CI Servers/Tools**

- Gitlab CI
- TeamCity
- Short introduction to Docker (optional)
- Running Jenkins with Docker
- Running Newman with Docker

## **Module 9: Workflows / Scenarios**

- Overview
- Creating a basic workflow / scenario
- Advanced workflows

## **Module 10: Data-Driven Tests**

- Using workflows to create data-sets
- Using external data files - Basic usage

- Using external data files - Writing tests
- Using external data files - Advanced usage

## **Module 11: Team Collaboration**

- Introduction
- Using the Git version control system - Overview
- Using the Git version control system - Hands on
- Team Workspaces - Creating and sharing a workspace
- Team Workspaces - Understanding users and permissions
- Team Workspaces - Managing user permissions

## **Module 12: Mock Servers**

- Introduction to mock servers
- Why use a mock server?
- Create your first mock server
- Recording responses from an existing API
- Known Limitations

## **Module 13: File Uploads**

- Sending and testing multipart/form-data requests (file upload form)
- Automating the upload and testing process with Newman

## **Module 14: Authentication / Authorization**

- Basic access authentication
- OAuth2 Authorization Flow (Authorization Code Grant)
- Form-based / Session-based authentication
- JSON Web Tokens (JWT)

## **Module 15: New Features in Postman**

- Designing APIs with Postman (OpenAPI 3.0.0 / Swagger)
- Sending GraphQL Queries in Postman

## **Module 16: JavaScript Fundamentals**

- Overview
- Data Types in JS
- Variables
- Conditionals
- Functions
- Data Structures – Arrays/Objects
- Loops
- Modern JS
- Apply what you have learn in Postman

- Conclusion

## **Course 9: AWS Cloud Certification Training**

### **Course Description:**

AWS Certified Solutions Architect is among the most valuable and highly sought-after cloud computing certifications in the world today. We designed this cloud architect certification training for anyone seeking to learn the major components of Amazon Web Services (AWS). By the end of the course, you'll be prepared to pass the associate-level AWS Certified Solutions Architect certification exam.

In this module, you will learn about the introduction to compute offering from AWS called EC2. We will cover different instance types and Amazon AMIs. A demo on launching an AWS EC2 instance, connect with an instance and hosting a website on AWS EC2 instance. We will also cover EBS storage Architecture (AWS persistent storage) and the concepts of AMI and snapshots.

- Introduction to Cloud Computing & AWS
- Elastic Compute and Storage Volumes
- Load Balancing, Autoscaling and DNS
- Virtual Private Cloud
- Storage - Simple Storage Service (S3)
- Databases and In-Memory DataStores
- Management and Application Services
- Access Management and Monitoring Services
- Automation and Configuration management
- AWS Migration
- AWS Architect Interview Questions

### **Placement Guide**

- What is an Interview?
- Tips to clear an Interview
- Common Interview questions and answers
- Software Testing Interview Questions and Answers
- Resume Building Guide
- Attempt for Software Testing Global Certification Exam
- Start applying for Jobs