



# Unit Testing Frameworks

**Duration:** 4 days

# Day 1:

Unit Testing Frameworks

* What are unit tests?
* The fundamentals of a unit testing framework

Test-Driven Development

* Refactoring overview
* The different approaches to testing

XUnit

* Writing and running a unit test
* Testing parameters with Inline
* Testing multiple parameters
* Using a custom class member

Advanced Optimizations

* Grouping tests
* Working with test attributes
* Customizing with playlists
* Debugging tests
* Using test menus
* Viewing results with CodeLens
* Using Code Coverage
* Running tests with MSTest

# Day 2:

Assertions

* Test Cases
* Test Classes
* Test Runners
* Ignoring Tests
* Initialization and Clean-up

Test Goals

* Boundary conditions





* Check Inverse Relationships
* Cross Checking Results
* Force Error Conditions
* Performance Characteristics

Jasmine.js

* Understanding Jasmine.js
* Writing specifications in Jasmine.js
* Learning built-in matchers
* Covering before and after

# Day 3:

Advanced

* Using Spy for Mocking
* Nesting describes blocks
* Writing Custom Matchers
* Asynchronous support using \*runs\*
* Testing UI using Jasmine.js
* Jasmine jQuery Helpers

Continuous Integration

* Introduction to Grunt.js
* Overview of Grunt tasks
* Creating and configuring a CI workflow using Grunt.js

Other

* Use Karma for browser testing
* Client-side dependency management using Bower.js

Mocks

* Angular Mocks library
* Dependency Injection
* Fake HTTP Back-end Requests with ‘$HTTP Backend’
* Jasmine Spies

Writing Effective Unit Tests

* Controllers
* Filters
* Directives
* Services Factories
* Templates & Routes





* Events
* Karma Automated Tests

Code Coverage

* Code Coverage
* Block Coverage
* Branch Coverage
* Line Coverage
* Collecting Coverage Data
* Selecting Coverage Units
* Coverage Visualization
* Coverage Analysis
* Exclusions

# Day 4:

Isolation Techniques

* Design for Test
* Private Accessors
* Test Instance
* Configuration Files
* Test Instance – Database
* Test Instance – Service
* Stubs

Microsoft Fakes

* Isolation Tool
* Adding Fakes
* Stub Classes
* Stub Methods
* Stub Properties
* Shims
* Shims Context
* Classes
* Methods
* Default Behavior
* Global Shims

JMeter

* Introduction



* Overview of Performance Testing
* Performance Testing Concepts
* Performance Testing Types
* Why to use performance Testing tool?
* What is JMeter?

Running JMeter

* JMeter's Class path
* Using a Proxy Server
* Non-GUI Mode
* Distributed Mode

Introduction to Elements of JMeter Test Plan

* Thread Group
* Controllers
* Samplers
* Logic Controllers
* Listeners
* Timers
* Assertions
* Configuration Elements
* Pre-Processor Elements
* Post-Processor Elements

Building a Test Plan

* Adding and Removing Elements
* Loading and Saving Elements
* Configuring Tree Elements
* Running a Test Plan

Adding Users

* Adding Default HTTP Request Properties
* Adding Cookie Support
* Adding HTTP Requests
* Adding Post-Processor for Correlation
* Adding a Listener to View/Store the Test Results
* Saving the Test Plan
* Running the Test Plan





Recording Tests Using JMeter

* Creation of Thread Group
* Adding HTTP Proxy Server
* Configuring HTTP proxy server
* Configuring the browser for recording the test script
* Capturing the test steps using JMeter

Handling the dynamic server values

* Handling User Sessions with URL Rewriting
* Using a Header Manager
* Handling the dynamic server values
* Parameterize the user sessions

Parameterize with test data

* Identifying the test data on AUT
* Open a csv file with JMeter
* Reading the data from CSV files
* Using the parameters in JMeter Tests

Adding Assertions to the test script

* Validating the response data related issues
* Validating the response size related issues
* Validating the threshold for the server response times
* Running the tests and analysing the Assertion results

Building a Monitor Test Plan

* Adding Server
* Adding HTTP Requests
* Adding Constant Timer
* Adding a Listener to View/Store the Test Results
* Adding Monitor Results
* Saving the Test Plan
* Running the Test Plan

Running Multiple Scripts with JMeter

* Creating multiple test thread groups
* Calling the multiple thread groups from a test plan
* Configuring the threads groups with respective users’ numbers and ratios
* Running the load test for multiple thread groups from a Single Test





* Analyze the user group specific issues
* Analyze the Test plan specific issues for all user groups

Reporting and Analyzing the Results

* Adding the Assertion Results and monitoring
* Configuring and Generating the Distribution Graph
* Configuring and Generating the Graph Full Results
* Generating the Monitor Results
* Saving the data through Simple Data Writer
* Configuring and Generating the Spline Visualizer
* Generating the Aggregate Graph for multi group test
* Generating the Aggregate Report for multi thread group test
* Generating the Summary Report

Best Practices

* Limit the Number of Threads
* Where to Put the Cookie Manager
* Where to Put the Authorization Manager
* Reducing resource requirements
* Bean Shell server
* Distributed Testing