



**TSL Course Contents**

**Course Outline**

**Day 1 - 5: (5 Days)**

**Oracle SQL & PL/SQL**

* Introduction to Databases.
* Database Models.
  + Relational Model
* Data Design and Normalization
* Structured Query Language and its categories
  + DDL – DML – DQL – DCL – TCL
* Oracle built in Functions
* SELECT statement varieties with clauses
  + WHERE clause
  + GROUP BY clause
  + HAVING clause
  + ORDER BY clause
* Retrieving data from multiple tables
* Joining the tables – Join variants
  + Equi and Non-Equi Joins
  + Self-Join
  + Cartesian Product
  + Outer Join with modern syntax
* Subqueries
  + Nested Subqueries
  + Co-related Subqueries
* Views

**DBMS – PL / SQL**

* PL/SQL Program Structure.
* Procedural programming using PL/SQL
  + Data Types and Scalar / Anchor Declarations
  + SELECT INTO statement
  + Use of CURSORS in PL/SQL
* Scenario Based Learning
* Modularized programming in PL/SQL
  + Anonymous v/s Named PL/SQL blocks
  + Stored Procedures
  + Functions
  + Packages
* Exception Handling In PL/SQL

**Day 6 - 20: (15 Days)**

**Introduction to Java**

* Different programming paradigms.
* Transformation from OO to PO Paradigm.
* Java Architecture
* A JAVA Program – General Strategy
* Source Code – Byte Code – Path - Class path
* Java – Execution Level Essentials - JDK – JRE – JVM
* Class Loaders and JVM Architecture

**Naming Convention, Data Types & Variables**

* Primitive datatypes, declarations
* Variable names, naming convention
* Numeric literals, character literals
* String, string literals
* Non-primitive datatypes, dot operator

**Operators & Expressions**

* Expressions
* Assignment operator
* Arithmetic operators
* Relational operators
* Logical operators
* Increment and decrement operators
* Operate-assign operators (+=, etc.)
* The conditional operator
* Implicit type conversions

**Control Flow Statements**

* Statements
* Conditional (If) Statements
* Adding An Else If
* Conditional (Switch) Statements
* While and do-while loops
* For loops
* A for loop diagram
* Enhanced for loop
* The continue statement
* The break statement

**Arrays**

* Introduction of arrays
* Types of arrays
* Operation with arrays

**Wrapper Classes**

* Introduction to wrapper classes
* Use of wrapper classes

**Object-Oriented Programming**

* Introduction to object-oriented programming
* Classes and objects
* Fields and methods
* Encapsulation
* Abstraction
* Inheritance
* Polymorphism and Polymorphic Behavior
  + Static Binding
  + Dynamic Binding

**Objects & Classes**

* Defining a class
* Creating an object
* Instance data and class data
* Methods
* Constructors
* Access modifiers

**Java Classes**

* Object class
* String class
* Stringbuilder and stringbuffer
* Other built-in classes

**Keyword & Packages**

* Static, final etc.
* Static imports
* Introduction to packages
* Defining packages
* Package scope

**Input/Output Streams**

* Overview of streams
* Bytes vs. Characters
* Binary input and output
* Printwriter class
* Reading and writing objects
* File IO

**Inheritance in Java**

* Inheritance in java
* Types of inheritance
* Method overriding
* This and super

**Interfaces & Abstract Classes**

* Introduction to interface
* Implementing and extending interfaces
* Multiple inheritance using interface
* Abstract classes
* Abstract class vs. Interface

**Inner Classes**

* Inner classes
* Member classes
* Local classes
* Anonymous classes
* Instance initializers
* Static nested classes

**Core Java – Exception Handling**

* What is Exception? Java’s exception handling mechanism.
* try – catch block
* finally block
* Exception Propagation and throw keyword
* throw v/s throws
* Exception handling with method overriding
* Custom Exceptions
* Scenario based learning

**Core Java – Threads and Concurrency**

* Multithreaded programming / Lifecycle of Thread.
* Thread Management
* Multithreaded programming
* Extending Thread class – Implementing Runnable Interface
* Synchronization and Thread Safety

**Core Java – Collections and Generics**

* Collections framework.
* ArrayList – LinkedList – HashMap – HashTable
* Other classes and interfaces in Collection Framework
* Comparable v/s Comparator
* Generic Collection

**Core Java – Annotations and Reflections**

* Annotations in Java.
  + Levels of annotations
* Built-In Annotations – Custom Annotations
* Reflection API
* Modifying the run time behavior of a class

**Advance Java - Java Database Connectivity**

* JDBC Introduction - JDBC architecture - JDBC driver types.
* JDBC API and various interfaces
  + Connection Interface
  + Statement Interface
  + ResultSet Interface
* Steps to establish a JDBC Connection
* Handling parameterized queries using PreparedStatement Interface
* Embedding PL/SQL objects using CallableStatement interface
* Transaction Management in JDBC

**Core Java – Mini Project**

**Basic of HTML, CSS & JavaScript**

**Online Assessment on Core Java & SQL, PL/SQL**

**Day 21 - 25: (5 Days)**

**Manual Testing - ISTQB**

**Day 26 – 35: (10 Days)**

**Introduction to Selenium**

* Selenium overview
* Selenium Fundamentals
* Selenium components - IDE, Server, Grid
* Basic structure of an IDE Test as HTML Table
* Selenium IDE Test Cases and Test Suites   
  **Exercises and Group test automation project execution**
* Concept of DOM and Webtables
* Locator and DOM related tools
* Selenium Locator Strategies
* Using Regular expressions in tests   
  **Exercises and Group test automation project execution**
* Actions, Assertions and Accesors in Selenese
* Handling JavaScript alerts, confirmation and prompts
* *Handling alerts, model windows and iframes*
* Handling AJAX and Test Synchronization
* Structure dependant and attributes based locators   
  **Exercises and Group test automation project execution**
* Selenium Servers
* Concept of WebDriver, WebDriverBacked and RC
* Difference Between RC and WebDriver
* Using WebDriverBacked from Selenium migration projects
* Installing and Configuring Selenium WebDriver in Eclipse
* Group test automation project execution
* Testing Frameworks for Selenium
* What is Junit and TestNG? Similarities and Differences
* Junit Test Suites
* TestNG Test Suites
* *Using TestNG to*

*1) Selectively run tests,*

*2) Run tests in parallel on different browsers*

*3) Grouping & prioritizing tests*

*4) Using data provider with excel integration to read data for data parameters in tests*

* Test Fixtures and Annotations
* Taking screenshots and trigger on error

**Exercises and Group test automation project execution**

* WebDriver in various scenarios
* Handling common objects such as Checkbox, Radio buttons, Dropdown, Autocomplete
* Dealing with mouse hover in WebDriver
* Controlling browser attributes and navigation

**Exercise and Group test automation project execution**

* Validating on CSS attributes of objects
* Handling Cookies, Find broken images/links in website
* Reporting the test execution (TestNG Report, Allure Report)
* *Using Extent reports for reporting results including writing test step results to the report*
* JavaScript action for scroll

**Exercise and Group test automation project execution**

* Implicit, Explicit and Fluent wait
* Various flavors Chrome, IE and Firefox
* WebDriver user defined actions and keyboard entries

**Exercise and Group test automation project execution**

* Data Driven frameworks for Excel
* Headless execution of Tests
* Reading test related variables from properties file

**Exercise and Group test automation project execution**

* Data Driven using CSV and Databases
* *Page object model framework*
* Page Factory and POM for better maintenance and reusability
* Combing Page Factory with Data driven test
* Updating the existing tests in POM model

**Exercise and Group test automation project execution**

* Selenium Grid
* Setup the Hub server
* Register the WebDriver Nodes and legacy RC nodes to the Hub server
* Parameterize the tests to run on various nodes
* Override the default parameters on the nodes
* CP-SAT Mock Exam

**Exercise and Group test automation project execution**

* Selenium in CI/CD
* Automating the Automation
* Understanding Testing tasks in Jenkins
* Installing Maven and dependencies using pom.xml
* Using Maven for Jenkins Selenium testing tasks
* Discussion on Mock Exam submissions to help improve on practical aspects

**Exercise and Group test automation project execution**

* Hybrid Framework concepts
* Difference between Keyword driven and Hybrid Frameworks
* Java Reflection Class
* Reporting results back to the framework

**Exercise and Group test automation project execution**

**Online Assessment on Manual Testing & Selenium**

**Day 36 - 43: (8 Days)**

**Project Simulation**

Writing Test Scenarios and automating using selenium.

**Project Evaluation – TSL SME’s:**

**2 Days Buffer for Certification**