SELENIUM TESTING COURSE CONTENT

PRE-SELENIUM

Core Java Programming
- Introduction to Java
- History of java
- Comparison with C and C++
- Features of Java
- JDK,JRE,JVM overview
- JDK Directory Structure
- Basic Java Program through command prompt

Installation and Setup:
- Download and install JDK/JRE
- Set Environment variables
- Download Eclipse IDE
- Coding standards followed in Eclipse
- Naming standards followed in Eclipse
- Features of Eclipse IDE

Packages
- Introduction to packages
- Need for packages
- package declaration in Java
- Import statement in Java
- static import in java
- Resolving name clashes in packages

OOPS and its application in Java:
- Classes and Objects
- Defining a class;Defining instance variables and methods
- Defining a class, variable and method in Java
- Method Signature; method call
- Creating objects out of a class
- Method calls via object references

Abstraction
- Interfaces and Abstract classes
• Abstract and non-abstract methods

Inheritance
• extends and implements keywords in Java
• Super class and Sub class
• this keyword, super keyword in Java for inheritance
• Concrete classes in Java
• Aggregation and Association

Polymorphism:
• Compile time polymorphism -- Overloading of methods
• Run time polymorphism -- Overriding of methods
• Method Overriding rules and method overloading rules
• Introduction to Object class and it's methods

Encapsulation:
• Protection of data
• Java Bean, POJO
• Getters/Setters
• Memory management in Java
• Heap
• Stac
• Garbage Collection

Data types:
• Primitive Datatypes
• Datatype Declarations
• Datatype Ranges and its calculation
• Memory allocation for each Datatype
• Variable Names Conventions
• Numeric Literals, Character Literals
• String Literals
• Arrays
• Array of Object References
• Enumerated Data Types
• Non-Primitive Datatypes

Operators:
• Expressions in Java
• Assignment Operator
• Arithmetic Operators
• Relational Operators
• Logical Operators
• Conditional Operators
• Operator Precedence
• Implicit Type Conversions
• Upcasting and downcasting
• Strict typing
• Type conversion

**Conditional Control Statements**
• Flowchart for conditional statements
• If statement
• If-else statement
• If-else-if statement
• Switch statement
• String in switch case

**Looping Control Statements**
• For loop
• While loop
• Do-while loop
• Unconditional Control Statements/Jump Statements
• break statement
• labelled break statement
• return statement
• continue statement

**Scanner class:**
• Diff between Scanner and BufferedReader
• Methods to get Primitive datatypes
• match method
• delimiter method
• findInLine method
• skip,close method
• useRadix method
• useLocale method
• IOException method
De-bugging:
• Launching and debugging java code
• Breakpoints
• Debug perspective
• Stepping commands
• Trace point, Trigger point
• Breakpoints grouping
• Breakpoints sorting

Access Modifiers in Java
• Role of access modifiers
• Private access modifier
• Role of private constructor
• Default access modifier
• Protected access modifier
• Public access modifier
• Access Modifier with Method Overriding

Types of variable:
• variable
• variable memory storage
• Static variable
• Local variable
• Global/Instance Variable
• variable widening
• variable narrowing

Constructor:
• Constructor
• Default constructor
• Non-arg based constructor
• Parameterised constructor
• Difference between Constructor and Method
• Constructor chaining
• this and super method
• constructor overloading

Singleton class:
• Singleton class
• Normal class vs singleton class
- Use of Singleton class
- JDBC Using Model Object and Singleton Class
- Collections.singleton() method in Java
- Private Constructors and Singleton Classes in Java
- Java Singleton Design Pattern Practices

**Factory Design Pattern:**
- Factory Design Pattern:
- Advantage of Factory Design Pattern:
- Implementing Factory Design Pattern:
- Abstract Factory Design Pattern
- Overview of other creational design pattern
- Builder design pattern

**String:**
- String datatype
- String declaration
- String Tokenizer
- String methods
- String types
- String memory allocation
- Manipulations in string
- Interfaces and classes in String

**Arrays:**
- Declaration
- Instantiation
- Initialization of Java Array
- Single dimensional Array
- Multi-dimensional Array
- Anonymous Array
- Cloning an Array

**Wrapper class:**
- Need of Wrapper classes
- Autoboxing
- Unboxing
- Primitive Wrapper Classes
• Utility methods of Wrapper classes
• `valueOf` and `xxxValue` methods
• `parseXxx` and `toString` methods

**Generics:**
• Need for Generics
• How Generics works in Java
• Types of Generics
• Generic Type Class or Interface
• Generic Type Method or Constructor
• Generic Type Arrays
• Generics with Wildcards
• Unbounded Wildcards
• Bounded Wildcards

**Collection:**
• Java Collection Framework
• Hierarchy of Collection Framework
• Collection interface
• Iterator interface
• Methods of collection interface
• List
• Set
• Queue
• Collections utility class

**Map:**
• Introduction to Map interface
• Methods in Map
• Iterating a Map
• Map hierarchy
• Sorted Map
• `LinkedHashMap`
• TreeMap
• HashMap

**Exception handling:**
• Exception types
• Usage of Try
• Usage of Catch
• Usage of Throw
• Usage of Throws
• Usage of Finally
• Built-in Exceptions,
• Creating own Exception classes

**Java Regex:**
• Regular expression
• MatchResult interface
• Matcher class
• Pattern class
• PatternSyntaxException class
• Regex Quantifiers
• Regular Expression Character classes
• Regex Metacharacters

**File operations:**
• File Handling in Java
• Stream
• Java File Methods
• File Operations in Java
• FileReader
• FileWriter
• BufferedReader
• File permissions

**Date and Time:**
• Method & Description
• Date Comparison
• Date Formatting Using SimpleDateFormat
• Simple Date Format Format Codes
• Date Formatting Using printf
• Date and Time Conversion Characters
• Sleeping for a While
• GregorianCalendar Class

**Serialization:**
• Serialization in Java
• Need for Serialization in Java
• Serializing an Object
• Deserializing an object
• Advantages and Disadvantages of Serialization in Java
• Practical examples of Serialization in Java
• Externalizable Interface
• Transient Keyword
• Serial Version UID
• Controversies of Serialization in Java
• Best Practices while using Serialization in Java

XML:
• Introduction to XML
• Read XML File in Java
• Java DOM Parser
• Java SAX Parser
• Read XML File in Java Using eclipse
• Reading XML file using DOM Parser
• Reading XML file using SAX parser

JSON:
• JSONArray
• JSONParser
• JSONObject
• Json.simple maven dependency
• Write JSON to file with json-simple
• Read JSON from file with json-simple
• Download Sourcecode

CSV:
• Read CSV File in Java
• CSV file creation
• How to create CSV File
• Java Scanner class
• Java String.split() method
• Using OpenCSV API
• Reading CSV file with a different separator

Multi-threading:
• Concepts ofThread
• Thread life cycle
- Creating threads using Thread class and Runnable interface
- Synchronization
- Thread priorities
- Inter Thread communication.

**jdk 1.5 features:**
- Autoboxing
- Generics
- Enhanced for loop
- Varargs
- Enums
- Static imports
- C-lang printf()
- StringBuilder
- Metadata

**jdk 1.7 features:**
- String in Switch Expression.
- Underscores Between Digits in Numeric Literals.
- Integral Types as Binary Literals.
- Handling multiple exceptions in a single catch block.
- Try-with-resources Statement.
- Automatic Type Inference in Generic object instantiation.

**jdk 1.8 features:**
- Lambda Expression
- Method references
- Functional interfaces
- Interface changes: Default and static methods
- Streams
- Stream filter
- forEach()
- Collectors class with example
- StringJoiner class with example
- Optional class with example
- Arrays Parallel Sort

**Memory management:**
- JVM Memory Structure
- Heap area
• Method Area
• JVM Stacks
• Native method Stacks
• Program counter (PC) registers
• Working of Garbage Collector
• Memory leaks in Java

SQL:
• Introduction to SQL
• Table creation
• SQL Insert
• SQL Update
• Applying Constraints
• SQL Syntax
• SQL Data Types
• SQL Operators
• SQL Database
• SQL Select
• SQL Clause
• SQL Delete
• SQL Join
• SQL Keys

JDBC Connection:
• Establishing connection
• Types of JDBC driver
• JDBC-ODBC Bridge Driver,
• Native Driver,
• Network Protocol Driver, and
• Thin Driver
• Running query
• Extracting Result

Java programs:
• Check the given number is odd or not
• Check the given number is even or not
• Print first 100 odd numbers
• Print first 100 even numbers
• Count the number of even numbers from 1 to 100
• Count the number of odd numbers from 1 to 100
• Find the factorial of a given number
• Generating fibonacci series
• Find the reverse of the given number
• Check the given number is palindrome or not
• Check the given number is armstrong or not
• Find the sum of the digits in a number
• Find the number of digits in a number
• Find the product of digits in a number
• Find the reverse of the string
• Check the given string is palindrome or not
• Print each word's first letter of the given string in capital number
• Print the duplicate numbers in array
• Print the Unique elements in array
• Remove the duplicate character in string
• Remove the duplicate words in string
• Write code to print patterns
• Pre-increment post increment example
• prime number or not
• Anagram or not
• Usage of Collections.min(),max() and sort()
• Usage of Arrays.min(),max() and sort()
• Print numbers as String

Coding standards:
• Coding Standards for Classes
• Coding Standards for Interface
• Coding Standards for Methods
• Coding Standards for Variables
• Coding Standards for Constants
• Java Bean Coding Standards
• Getter Methods
• Setter Methods

SELENIUM:

Selenium Introduction
• Types of Applications (Desktop, Web, Mobile, Hybrid)
• Software Testing Methods (Manual and Test Automation).
• Selenium Introduction
• Selenium Components
• Selenium vs. Other Testing Tools
• Advantages of Selenium
• Integration of Selenium with Other Tools

**Selenium Components:**
• Purposes and functionalities
• Understanding the components
• Selenium RC
• Selenium IDE
• Selenium webdriver
• Selenium Grid
• When to use Grid

**WebDriver**
• Third party drivers and plugins
• Driver requirements
• What is WebDriver
• Selenium Architecture
• Simple Program in Selenium WebDriver
• WebDriver methods

**Types Of Browser Launch**
• Desired Capability
• Downloading driver file
• Downloading selenium jarfile
• Chrome Browser Launching
• Safari Browser Launching
• InternetExplorer Browser Launching
• Installing FireBug and FirePath
• Firefox Browser Launching

**Locators**
• Inspecting elements in different browsers.
• id
• name
• classname
• xpath
• tagName
• linkText
• partiallyLinkText
- cssSelector

Xpath
- Contains Xpath
- Text Xpath
- Text Contains Xpath
- Attribute with contains
- Following
- Ancestor
- Child
- Preceding
- Following-sibling
- Parent
- Self
- Descendant

Types Of Xpath
- Relative Xpath
- Absolute Xpath
- Difference between Absolute Xpath and Relative Xpath
- Limitations in Absolute xpath
- Advantages of using Relative xpath

Check Box
- Finding checkboxes count
- Checking the visibility of Check Box
- Checking the properties of Check Box
- Identifying common locator for all checkboxes
- Checking toggled attribute

Text Box
- Handling the Text Box
- Checking the visibility of Text Box
- Checking the properties of Text Box
- Identifying common locator for all TextBoxes
- Finding Textboxes count

Radio Button
- Handling the Radio Button
- Checking the visibility of Radio Button
• Checking the properties of Radio Button
• Identifying common locator for all Radiobuttons
• Finding radiobuttons count

**WebElements**
• What are WebElements in Selenium
• Different types of WebElements
• Operations performed on the WebElements
• How to locate the WebElements on the web page
• Different WebElement methods
• Difficulties while handling webElements

**Dynamic Locators**
• Absolute Path method
• Relative XPath method
• Identify by index
• Preceeding-sibling,Following-sibling concept
• Ancestor ,parent concept
• Common tagname(*) method
• Multiple attributes to locate an element

**Desired Capability**
• Need for Desired Capabilities
• Different types of Desired Capabilities Methods
• Example for set capability method
• Setting the Property
• Getting the Property

**Navigation Commands**
• Navigate To Command
• Forward Command
• Back Command
• Refresh Command
• navigate method over get method
• Navigation by using JavascriptExecutor

**WebDriver Commands**
• Fetching a web page
• Locating elements and sending user inputs
• Clearing User inputs
• Fetching data over any web element
• Performing Click event
• Navigating backward in browser history
• Navigating forward in browser history
• Refresh/Reload a web page
• Closing Windows
• Closing Browser
• Handling Windows
• Handling Frames
• Handling Drag and Drop

**Actions**

• Drag and Drop
• Mouseover Action
• Right Click
• Double Click
• Performing Multiple Actions
• Accessing modifier keys using Actions class

**Alerts**

• Switching into Alert
• Alert methods
• Types of Alert
• Handling the Alert
• Passing the inputs to Alerts
• Entering text into Alert
• Get the text present in Alert
• **Pop-ups**
• Handling the Window based popups
• Handling the Notification popups
• Handling pop-ups using Robot class
• Handling the Login popups
• Chrome Options
• FirefoxOptions
• InternetExplorerOptions

**Robot Class**

• Need of Robot Class
• Methods to implement this class
• Mouse click using Robot class
• Limitations
• Copy Operations
• Cut Operations
• Paste Operations
• File Uploading
• Alert Handling

Waits
• Need for Waits
• Static waits
• Dynamic waits
• Implicit Waits
• Explicit Waits
• Fluent Waits
• WebDriver Waits

JavaScript
• WebElement Highlighting
• Click Operation
• Fetching the Data from WebElement
• Sending the Inputs to WebElement
• Scrolling Operations
• Highlighting a WebElement

ScrollUp/ ScrollDown
• Scroll the web page by pixel
• Scroll the web page by the visibility of the element
• Scroll down the web page at the bottom of the page
• Horizontal scroll on the web page
• Multiple Scroll
• ScrollBy coordinates

Frames
• Need for Frames
• Identifying a Frame
• Switching to Frames using Selenium WebDriver
• Different ways of switching
• Dynamic frames handling
• Frames Size
• Concept of Nested Frames
Windows Handling
- Importance of Windows Handling
- Handling the Multiple Windows
- Windows Handling using Set
- Windows Handling using List

WebTable
- Analyzing WebTable structure in DOM
- Handling multiple webtables in a page
- Dynamically changing WebTable handling
- Extracting values from webTable
- Analyzing the Tagnames
- Different Scenarios with WebTable

Dynamic WebTable
- Handling Dynamic Tables In Selenium
- Analyzing the Dynamic WebTable
- Analyzing the HTML Tags in Dynamic WebTable
- Different Scenario with Dynamic WebTable

ScreenShot
- Need of Screenshot in Automation testing
- Capture Screenshot in Selenium
- Capture Full Page Screenshot
- Taking a Screenshot of a particular element of the page
- Taking a Screenshot with different file formats
- Random name generation for screenshots

Image
- Finding images count in webpage
- Finding broken images count in webpage
- Finding broken image URL
- JavaScriptExecutor code to verify if image
- code to print desired output as per image

Links
- Identifying URL
- Validating URL
- To Find a broken links
• HTTP response code
• Collect all the links in the web page

**DropDown**
• Select class in Selenium WebDriver
• Different Select commands
• Multiple Select commands
• DeSelect Commands
• Get All options
• Dropdown without Select tag
• Handling dropdown with values changing its position dynamically.

**File Upload/File Download**
• Uploading files in Selenium WebDriver using Sendkeys
• Uploading files in Selenium WebDriver using Robot Class
• Uploading files in Selenium WebDriver using AutoIT
• Download files in Selenium WebDriver using Sendkeys
• Download files in Selenium WebDriver using Robot Class
• Download files in Selenium WebDriver using AutoIT

**Auto IT**
• download and install AutoIT
• Finding element through element Identifier
• Writing script on AutoIT editor
• AutoIT Upload file in Selenium Webdriver

**Tooltip:**
• Advanced User Interactions API
• Get Tooltip Text in Selenium Webdriver
• Tooltip using the "title" attribute
• Tooltip using a jQuery plugin

**Browser Stack**
• Introduction to Browser Stack
• Cross Browser Testing
• BrowserStack History
• Features of BrowserStack
• Testing The Web Application
• Browser Stack Key Functions
• Testing The Mobile Application In Mobile Browsers
• Testing Of Native, Hybrid Mobile Application In BrowserStack

**Sauce Lab**
• Saucelab-Introduction
• Value Proposition
• Manual testing on Sauce labs
• Post Execution
• Automated Test Execution
• saucelabs gem
• Execution and Results

**POST SELENIUM**

**Maven**
• Introduction to Apache Maven
• Maven Dependencies
• Maven Plugins
• Controlling The Build
• Maven Release Process
• Deploying to a Repository
• Using Snapshots

**Git**
• Introduction To Git
• Working Locally With Git
• Working Remotely With Git
• Branching, Merging And Rebasin With Git
• Using The GitHub Website
• GitHub For Windows Basics
• Teamwork With GitHub For Windows
• Social Coding With GitHub

**Jenkins**
• Continuous Integration or Continuous Deployment concepts
• Installing and Configuring Jenkins
• Freestyle Project Configuration
• Jenkins Pipelines
• Testing With Jenkins
• Pipeline Enhancements o Multi-branch Pipelines and Code Promotion
QTTest
- Test plan: Releases and Builds
- Requirements
- Creating Testcases
- Creating and using parameters
- Using data query
- Executing test runs and submitting defects
- Complex scenario
- Setting up qTest explorer (web and desktop)
- using qTest Explorer (Web and Desktop)

ALM
- Introducing HP ALM
- Release Specifications
- Requirements Specifications module in HP ALM
- Test Plan Modules
- Defect Management Life Cycle
- Report & Analysis

Bamboo
- Getting started with Bamboo
- Understanding Bamboo
- Getting started with Java and Bamboo
- Getting started with .NET and Bamboo
- Configuring plans
- Deployment projects using Bamboo
- Evaluator
- Developer
- Getting support
- Automatic plan branches
- Quarantining intermittent tests
- Integrating Bamboo with Atlassian applications
- Bamboo remote agent installation guide
- Getting feedback
- Importing data from Jenkins
- Administering
- All administration topics
- Release notes
- Installing and upgrading
Ant
- Introduction to Ant
- Installing Ant
- Ant concepts and terminologies
- Working with Ant
- Integrating ANT into your IDE
- Ant in the Real-World
- Ant Tasks
- Automating your build and testing through Ant tasks
- Using Ant for automated deployment
- Integrating Ant with a version control system
- Creating Custom Ant tasks

BitBucket
- Introduction to Bitbucket
- Bitbucket Installation
- Navigation of Bitbucket
- Introduction to Git.
- Git installation
- Git commands.
- Git with Bitbucket
- Repository creation
- Repository permissions
- Branches
- Branches permission
- pull requests
- merging files
- Repository clone
- Git bash
- Source tree
- Installation of source tree
- User management
- Integration with jira

Gradle
- Gradle-plugins
- Working with files
- Ant Integration
- Dependency Management
- Extending the model
- Task inputs & outputs
- The Java plugin
- Multiproject builds
- The build runtime

**JIRA**
- Introduction To JIRA
- Test Management In JIRA (Zephyr)
- Advanced Search And Introduction To JQL (JIRA Query Language)
- Generating Reports In JIRA
- Introduction To JIRA Agile

**Data Driven:**
- Data Driven Explaination
- Excel Types
- Apache poi-ooxml
- Interface, class, methods in apache poi-ooxml
- Read Data from excel
- Write data in excel
- update data in excel
- Creation of Automation scripts reading data from excel

**POM Framework:**
- Page Object Model
- Advantages of POM
- Implementing POM
- Object repository creation
- Page Factory
- POM Annotation
- Getters and setters
- AjaxElementLocalFactory

**JUNIT Framework:**
- JUNIT Test Framework
- download and installation of junit
- JUNIT Annotation & API
- JUNIT Assert
- Create JUNIT Test Suite
- JUNIT Ignore Test
• JUNIT ErrorCollector
• JUNIT Parameterized Test
• JUNIT Vs Testng
• Execution metrics from JUNIT

TESTNG Framework:
• TESTNG with Selenium
• Advantages of TESTNG over JUnit
• TESTNG Annotation & API
• Create TESTNG Suite
• TESTNG Parameterized Test
• Parameters
• Data Provider
• Re-running failed Test Case
• IretryAnalyzer
• IAnnotationTransformer
• TESTNG Assert
• Hard Assert
• Soft Assert
• Run Multiple test case
• Customized, PDF, & Email Testng reports in selenium webDriver
• TESTNG Ignore Test

Cucumber FrameWork:
• BDD Introduction
• Cucumber Introduction
• How to download & Installation plugin
• Gherkin Language
• Feature files
• TestRunner with junit
• Step Definition
• Scenario and Scenario Outline
• BackGround and Hooks
• BaseClass and POM Integration
• CucumberOptions
• Reports
• Rerun failed scenarios
**JBehave Framework:**
- JBehave Introduction
- How to download & Installation plugin
- Keywords in JBehave
- Stories
- Annotations
- Runner class
- List, story, Node implementation
- Reports generation

**Keyword-driven framework**
- Introduction to keyword-driven framework
- Function Library.
- Excel Sheet To Store Keywords.
- Design Test Case Template.
- Object Repository for Locators.
- Test Scripts or Driver Script.
- Retrieving data using keywords

**Hybrid framework**
- Components of Hybrid Driven Framework
- Function Library
- Excel Sheet to store Keywords
- Design Test Case Template
- Object Repository for Elements/Locators
- Test Scripts or Driver Script

**Travel domain**
- Domain explanation
- Keywords explanation in travel domain
- Different Modules explanation in travel domain
- Booking Flow explanation
- Different interfaces integration explanation
- Test scenarios preparation
- Testcase preparation
- Defect raising using JIRA
- Automating e-travel site using framework

**e-commerce domain**
- Domain explanation
- Keywords explanation in e-commerce domain
- End to end flow explanation in e-commerce domain
- Different interfaces integration explanation
- Test scenarios preparation
- Testcase preparation
- Defect raising using JIRA
- Automating e-commerce site using framework

**Other Domains Covered**
- Banking Domain
- Retail Domain
- Insurance Domain
- Healthcare Domain
- Educational Domain

**Manual testing**
- Software build process
- SDLC
- STLC
- Introduction to Testing
- Manual and Automation Testing
- Testing Techniques
- Testing principles
- Test scenario and Testcases preparation
- Software testing methodologies
- Functional and non-functional testing
- Detailed explanation for Waterfall methodology
- Detailed explanation for Agile Methodology
- Defect raising using JIRA

**Agile methodologies**
- Agile Methodologies Overview
- Agile Scrum
- Agile ceremonies
- Agile artifacts
- Kanban
- Adaptive Project Framework (APF)
- Extreme Project Management (XPM)

**Appium testing**
- Mobile Testing - Overview
- Mobile Testing - Platforms
- Mobile Testing - Device Types
- Native Vs Hybrid Vs Mobileweb
- Appium Architecture
- Prerequisite to use APPIUM
- Install Appium Desktop
- APPIUM Inspector
- Attach Android Emulator to Appium
- ADB commands
- APPIUM Test Case for Native Android App
- Limitations using APPIUM
- Common Encountered Errors In Appium
- Troubleshooting Steps in Appium
- Hardware Perspective
- Mobile Device Testing - Types
- Mobile Testing - Framework Overview
- Mobile Testing - Android Frameworks
- Mobile Testing - IOS Frameworks

**API testing**
- Set-up of API Test environment
- Types of Output of an API
- Test Cases for API Testing
- Approach of API Testing
- Difference between API testing and Unit testing
- What to test for in API testing
- Best Practices of API Testing
- Types of Bugs that API Testing detects
- Tools for API Testing
- Challenges of API Testing

**SOAPUI**
- Introduction to SOAP UI
- Functional API testing
- Supported Protocols/Technologies
- SOAP-INTEGRATION with Other Automation Tools
- SOAP UI Vs Selenium:
- SOAP UI Vs SOAP UI PRO
- SOAP UI PRO Version
- SOAP UI – Version Timelines
Postman
- Postman Introduction
- Working with GET Requests
- Working with POST Requests
- Parameterize Requests
- Create Postman Tests
- Create Collections
- Run Collections using Collection Runner
- Run Collections using Newman

Understanding Real-time scenarios and work culture
- Project explanation
- Challenges faced by tester
- Roles and responsibilities of a tester
- Day to day activities of a tester

Resume preparation
- Your resume is arguably the most critical part of the modern job application process.
- A well-organized, tailored resume will increase your chances of landing an interview while a poorly written resume could get lost in the sea of applicants.
- We will help you to prepare your impressive resume with best specification of your skill set

Mock-Interview
- Mock interviews by Greens technologies gives you the platform to prepare, practice and experience the real-life job interview.
- Familiarizing yourself with the interview environment beforehand in a relaxed and stress-free environment gives you an edge over your peers.
- Our mock interviews will be conducted by industry experts with an average experience of 5+ years. So you’re sure to improve your chances of getting hired!

Real-Time project
- To put your knowledge on into action, you will be required to work on two industry-based projects that discuss significant real-time use cases.
- These projects are completely in-line with the modules mentioned in the curriculum and help you to clear the certification exam.