
CTR- 504 QA and automation with Python

Overview

Course Duration: 200 Hours

About This Course

No previous knowledge.

Audience Profile

For people that want an introduction to the software programming industry.

At Course Completion

At the end of the course the participants will be ready to find work as QA automation engineers.

Course outline

Introduction to Python:

- Overview of the language, installation, and basic syntax
- Variables, data types, and operators

Control Flow:

- Conditional statements (if/else)
- Loops (for/while)

Functions:

- Defining and calling functions
- Arguments and return values
- Scopes

Data Structures:

- Lists
- Tuples
- Dictionaries
- Sets

Modules and Packages:

- Importing and using external libraries
- Creating and publishing packages

File Input/Output:

- Reading and writing files
- Exception handling

Object-Oriented Programming:

- Classes and objects
- Inheritance and polymorphism

Advanced Topics (optional):

- Decorators
- Generators
- Context managers
- Regular expressions

Project:

- Designing and building a small project using the concepts learned throughout the course

Database Integration:

- Connecting to relational databases
- Performing CRUD operations using SQLAlchemy
- Using NoSQL databases like MongoDB

Web Development:

- Understanding web development concepts
- Creating dynamic web applications using Flask
- Serving and consuming RESTful APIs

Project:

- Designing and building a complex project using the advanced concepts learned throughout the course

Introduction to Selenium and Setup

- Introduction to Selenium and its uses
- Downloading and installing Selenium and necessary drivers
- Setting up your first Selenium test

Basic Selenium Commands

- Finding elements on a webpage using Selenium
- Interacting with elements (clicking buttons, filling out forms, etc.)
- Understanding different types of element locators

Advanced Selenium Commands

- Navigating between pages and managing multiple windows
- Handling alerts and pop-ups
- Working with iframes and frames

Testing Frameworks and Best Practices

- Overview of testing frameworks (pytest, unittest, etc.)
- Writing test cases using a testing framework
- Best practices for writing and organizing Selenium tests

Handling Dynamic Content and Wait Strategies

- Understanding dynamic content and how it affects your tests
- Using wait strategies to handle dynamic content
- Strategies for testing web applications with slow or unreliable loading times

Data-Driven Testing and Reporting

- Data-driven testing with Selenium
- Generating reports and logging test results
- Understanding and analyzing test results

Advanced Topics in Selenium

- Browser automation and mobile testing
- Integration with other tools (CI/CD pipelines, etc.)
- Performance testing with Selenium

Introduction to QA Documentation

- Overview of QA documentation and its importance in software development
- Different types of QA documents (test plans, test cases, bug reports, etc.)
- Common standards and templates for QA documentation

Test Planning

- Understanding the software development life cycle (SDLC) and the role of test planning
- Creating a test plan that includes scope, objectives, timelines, and resources
- Identifying and prioritizing testing types and techniques based on project requirements

Test Case Creation

- Understanding the purpose and structure of a test case
- Writing test cases that are clear, concise, and thorough
- Incorporating test data and expected results into test cases

Test Execution and Reporting

- Understanding the importance of test execution and reporting
- Documenting test results and issues using a bug tracking system
- Providing stakeholders with regular updates on testing progress and results

Documentation for Agile and DevOps

- Adapting QA documentation practices for Agile and DevOps methodologies
- Using tools such as Jira and Trello to manage test cases and track testing progress
- Creating documentation that supports continuous integration and continuous delivery (CI/CD) pipelines

Test Metrics and Analysis

- Understanding the importance of test metrics and how they can be used to improve testing
- Identifying and tracking key performance indicators (KPIs) for testing
- Using test analysis to identify areas of improvement and make data-driven decisions

Documentation Best Practices and Review

- Reviewing and refining existing QA documentation based on best practices and lessons learned
- Incorporating feedback from stakeholders and team members
- Ensuring documentation is accurate, up-to-date, and accessible to relevant parties