

# **STQA VIVA**

## **MODULE 1**

### **1. Study of Review:**

- Q1: What is the purpose of a code review?
  - A1: Code review is conducted to find and fix defects, ensure adherence to coding standards, and enhance overall code quality.

### **2. Construction of Control Flow Graph:**

- Q2: Explain the concept of a Control Flow Graph (CFG).
  - A2: A Control Flow Graph is a graphical representation of a program's control flow, illustrating the flow of control among various program statements.

### **3. Writing Test Cases:**

- Q3: Why is writing test cases important in the software development process?
  - A3: Writing test cases is crucial for systematically verifying that software functions as intended, validating requirements, and ensuring the identification of defects.

### **4. Unit Testing:**

- Q4: What is the main objective of unit testing?
  - A4: The main objective of unit testing is to validate that individual units or components of a software work as designed in isolation.

### **5. Integration Testing:**

- Q5: How does integration testing differ from unit testing?
  - A5: Integration testing focuses on verifying the interaction between different components or systems, ensuring they work together seamlessly, whereas unit testing tests individual components in isolation.

### **6. System Testing:**

- Q6: When is system testing typically performed in the software development lifecycle?
  - A6: System testing is usually performed after unit and integration testing, validating the entire software system in its integrated state.

**Test Case:**

A set of conditions or variables under which a tester will determine whether an application or system works correctly.

**Code Review:**

The process of examining code for defects, ensuring adherence to coding standards, and improving overall code quality.

**Control Flow Graph (CFG):**

A graphical representation of a program's control flow, depicting the flow of control among different program statements.

**Unit Testing:**

The process of testing individual units or components of a software in isolation to ensure they function as intended.

**Integration Testing:**

Verifying the interaction between different components or systems to ensure they work together seamlessly.

**System Testing:**

Testing the entire software system as a whole to validate that it meets specified requirements and functions correctly in its intended environment.

## **MODULE 2**

### **Automation Testing :**

Automation testing is the use of specialized tools and scripts to automate the execution of tests on software applications. It is employed to increase testing efficiency, coverage, and repeatability. Automation testing helps in executing repetitive test cases, regression testing, and achieving faster feedback.

### **Selenium :**

Selenium is an open-source automation testing framework primarily used for web applications. It supports various programming languages and browsers, providing flexibility for testers. Its popularity stems from its versatility, cross-browser compatibility, and strong community support.

### **Selenium WebDriver:**

The WebDriver component of Selenium allows interaction with web elements and browsers to automate testing.

### **Browser Driver:**

A software component that allows Selenium WebDriver to communicate with and control a specific web browser (e.g., ChromeDriver for Google Chrome).

### **Test Script:**

A set of instructions written in a programming language to automate the execution of test cases.

### **Cross-Browser Compatibility:**

The ability of a web application to function correctly across different web browsers.

## MODULE 3

### WebDriver:

WebDriver is a tool in Selenium used for automating interactions with web browsers. It provides a programming interface to control browser actions.

### Locators:

Locators are mechanisms used by Selenium to identify and interact with web elements on a page. Common locators include ID, CSS selectors, and XPath.

### Synchronization:

Synchronization in Selenium manages timing differences between script execution speed and webpage element loading. Implicit waits, explicit waits, and fluent waits are used for synchronization.

### Alerts:

Alerts are pop-up boxes on a webpage requiring user interaction or displaying important messages. Selenium provides the `Alert` interface for handling alerts.

### Action Classes:

Action classes in Selenium, like `Actions`, allow performing complex user interactions such as mouse movements and keyboard actions.

### Dropdowns:

Dropdowns are UI elements allowing users to select an option from a list. Selenium uses the `Select` class to interact with dropdowns.

### Waits:

Waits in Selenium are used to pause script execution until a specific condition is met. Implicit waits, explicit waits, and fluent waits are commonly employed.

### Handling Frames:

Frames in Selenium are managed using the `switchTo().frame()` method, enabling interaction with elements within frames.

## **Locators in Selenium:**

Common locators include ID, CSS selectors, and XPath, providing ways to identify and interact with web elements.

## **Handling Alerts:**

Alerts in Selenium are handled using the `Alert` interface, allowing switching to alerts, accepting, dismissing, or extracting text.

## **Action Classes in Selenium:**

The `Actions` class is employed for complex user interactions, including mouse movements, keyboard actions, and drag-and-drop operations.

## **Dropdowns and Select Class:**

Dropdowns are interacted with using the `Select` class, offering methods to select options by index, value, or visible text.

## **Waits in Selenium:**

Waits, including implicit waits, explicit waits, and fluent waits, are used to manage timing issues, ensuring scripts wait for specific conditions before proceeding.

## MODULE 4

### TestNG (Test Next Generation):

TestNG is a testing framework inspired by JUnit and NUnit, designed to simplify testing activities such as configuration, parallel execution, and data-driven testing in Java applications.

### Installing TestNG:

TestNG can be installed in different ways, including using Maven, Gradle, or directly adding TestNG to the project's classpath. IDEs like Eclipse and IntelliJ also provide plugins for easy TestNG integration.

### TestNG Test:

A TestNG test is a method annotated with `@Test` that represents an individual test case. TestNG tests can have various annotations for configuration and control of test execution.

### TestNG Annotation:

Annotations in TestNG provide metadata to methods, influencing how TestNG treats them during test execution. Examples include `@Test`, `@BeforeSuite`, `@AfterSuite`, `@DataProvider`, etc.

### Testing .xml:

TestNG uses XML files for test suite configuration. The testing XML file specifies test classes, methods, parallel execution settings, and suite-level configurations.

What are some commonly used TestNG annotations?

- Answer: Common TestNG annotations include `@BeforeSuite`, `@AfterSuite`, `@BeforeTest`, `@AfterTest`, `@BeforeClass`, `@AfterClass`, `@BeforeMethod`, `@AfterMethod`, and `@DataProvider`.

## **MODULE 5**

### **Basic Types:**

Definition: Basic types in programming refer to fundamental data types, such as integers, floats, characters, and booleans, that are used to represent and manipulate data.

### **Linear Scripting:**

Definition: Linear scripting is a simple testing approach where test scripts are written in a sequential manner, executing commands one after another without branching or looping structures.

### **Library Architecture Framework:**

Definition: A library architecture framework is a testing framework where common functions and reusable code are organized into libraries. Test scripts use these libraries to perform actions, promoting code reuse and maintainability.

### **Data-Driven Framework:**

Definition: A data-driven framework is a testing framework where test data is separated from the test script, allowing the same script to be executed with different sets of data. It enhances flexibility and reduces the need for script modification.

## **MODULE 6**

### **Software Quality Assurance (SQA):**

Definition: Software Quality Assurance is a systematic process to ensure that software development and testing processes are conducted in a standardized and effective way. It involves establishing and adhering to quality standards throughout the software development life cycle.

### **Validation Checks:**

Definition: Validation checks are procedures or tests carried out to confirm that a system or software product meets specified requirements. This ensures that the product satisfies the intended use and produces the expected outcomes.

### **Can you provide an example of a validation check in software testing?**

- Answer: An example of a validation check is verifying that a login functionality allows only authorized users to access the system. This ensures that the software validates user credentials correctly.