Learning Objectives of CP-SAT v 2.0

"Knowledge with experience is power; certification is just a by-product"

Table of Contents

What is CP-SAT?

CP-SAT stands for "Certified Professional – Selenium Automation Testing" certification prepared and honored by "Agile Testing Alliance"

- CP-SAT certification exam can be taken by anyone without going through the training program covering the learning objectives below.

- CP-SAT certification exam is the only globally accepted certification exam covering practical assessment.

- CP-SAT learning objective gives the coverage topics for anyone who want to give the exam. Any training covering the CP-SAT Syllabus would be covering the learning objectives in a practical manner.

- CP-SAT exam can be taken in three language streams

  - Java
  - C#
  - Python

If you want to be accredited license trainer or partner get in touch with us at CPSATSupport@ATAEvents.org

How is it useful?

CP-SAT is designed to train testing professionals with the concepts of testing web applications using Selenium.

These learning objectives focus on the most challenging areas for testing web applications using the **latest Selenium 3.x**, using JUnit and TestNG. There is an extensive coverage on **Data Driven Testing**, Page Object Model, Selenium Grid, Continuous Integration (CI) with Jenkins and Maven. We have added GIT (SCM tool) integration with Jenkins and basics of Docker. How to use Selenium Grid on cloud is a new addition to the program

Learning and getting practically assessed on the most relevant topics or Selenium would be the key take aways.

Learning Objective Version tracking

What has changed in 2.0 Version?

- CPSAT Java certification program requires participants to be familiar with the java programming language constructs. We have added a pre-requisite section for Java
- Selenium IDE has been re-introduced
- Tools for locator creation – developer tools from Chrome and/or Selectors hub reference is now added
- Selenium grid execution using docker containers has been added
- Managing project through GIT and Continuous testing using GIT and Jenkins has been added
- Reporting section has been added
- Junit 5 has been added
- Handling CSV and Json files for Data Driven testing has been added
- Keyword driven framework <u>has been removed</u>

What had changed in 1.31 version?

- Starting with this version CP-SAT is now available in three different language streams- 1. Java 2. C# and 3. Python
- In Java stream - Use of JXL library for Data Driven Testing has been removed. Participants can use the library but it is not mandatory to know the library from this version onwards
- Reference to some theoretical concepts including WebDriver being flexible and having a lightweight API has been removed from the section "Running WebDriver tests section".

What had changed in 1.3 version?

- ANT as a build tool has been removed
- Headless browser execution is included
- Maven now is the choice of build tool
- Helper Utility class has now been included in the framework section
- Page object Model is given more focus and time
- Data Driven Framework Using JXL and POI library has been given more focus
- IDE and RC/Webdriver Backed references have been completely withdrawn

Eligibility, Exam duration and Enrollment

1. **Am I Eligible?**
   Anyone having basic experience in Java programming and manual testing can go for this certification.

2. **Duration of exam ?**
   CP-SAT exam has two parts, theory exam is 60 minutes. Practical exam is 150 minutes. Pass percentage of 60% is required in both the exams to qualify for CP-SAT certificate.

3. **How do I enroll myself?**
   You can enroll yourself by writing an email to
   CPSATSupport@ATAEvents.org

   Or

   You can visit the CP-SAT website.
   https://cpsat.agiletestingalliance.org/ and register for the program or the exam.

Prerequisites for CP-SAT (Java)

Java is a pre-requisite for attending CP-SAT training program. Please look at the following topics that would help participants in the CP-SAT Java learnings.

Agile Testing Alliance continuously runs Java Boot camps, the manuals are available on the CPSAT download section. URL is as below.

https://cpsat.agiletestingalliance.org/downloads

1. Learn Java as a programming language
   a. Syntax overview
   b. Data types in Java
   c. Operators in Java
   d. Collections in Java – Arrays, HashMap, ArrayList
   e. Strings
   f. File handling – reading and writing (CSV, Excel and Json files)
   g. Errors and Exceptions
   h. OOPs – Classes, Objects, Inheritance, Interfaces, Packages
   i. Static methods and member variables

Learning Objectives for CP-SAT Java Stream

Below learning objectives / concepts must be practiced throughout the program using practical case studies.

1. Tool background and Initial Setup

o   **Selenium Overview**

Learn about the definition and usage of the tool. Get aware of the cross-language, cross-platform, cross-browser capabilities of the tool. Learn about the Selenium tool license policy. Elaborate on the two parts of the Selenium toolkit. Introduction to Selenium IDE, WebDriver and Selenium Grid. Learn the importance of test automation in the Agile and Continuous testing environment.

o   **Tool Setup**

- Learn how to setup Eclipse and Maven
    - o Installation of eclipse and creation of workspace
    - o Importing Maven projects
    - o Learn about dependencies of the Maven goals
    - o Installation of TestNG eclipse plugin

- GIT setup (SCM Tool)

2. Selenium IDE
Learn about
- o Record and Playback
- o Adding of validations

2. Locators
- Walk through of Web Technologies
    - o HTML
    - o XML
    - o DOM
    - o CSS
- Learn about the Structure Dependent locators:
    - o XPath
    - o CSS
- Learn about the Attributes-based locators:
    - o Id
    - o Name
    - o Link Text

- Partial Link Text
- Tag Name
- Class Name
- Learn about the tools for locator formulation
  - Chrome developer tool and / or SelectorsHub

3. Selenium

3.1. Setting up Selenium WebDrivers for various Browsers

- Firefox Driver (Gecko driver)
- Chrome Driver
- Edge Driver

3.2. Headless browser Testing and driver capabilities
Learn about
- Headless Browser Testing
- Learn to manage driver capabilities

3.3. Fundamental Pillars (By, WebElement, FindElement, FindElements etc.)

Learn about
  - By
  - WebElement
  - FindElement
  - FindElements
  - Window switching and navigation
  - How to handle form elements
  - Keyboard user interactions
  - Capturing Screenshots

3.4. Handling Frames, Popups and Alerts
Learn how to handle
  - Frames
  - Alerts
  - Popups

3.5. Synchronization
Learn how important it is to use asynchronous waits rather than hard coded waits.

Learn about
  - Implicit wait
  - Explicit waits using

- Wait.Until
- ExpectedConditions

3.6. Working with Advance html elements (Tables, Dynamic Tables, paginations, Context menus, mouseover, drag and drop, Dropdowns, Dynamic lists.

- Tables
- Dynamic Tables and Pagination
- Using Action class (drag and drop, context click, mouse over)
- Using Select class for Drop Downs
- Handling Dynamic Lists / JQuery (Auto Complete)

3.7. Working with Java Script Executor and Handling exceptions

Learn how to scroll on the page and how to handle exceptions like stale element exception or element not interactable exception.
- Scrolling using Java Script Executor
- Clicking for hidden elements using Java Script Executor
- Understanding and handling stale element exception

4. Unit Test Framework(s)– TestNG and Junit 5

- a. Test Case and Test Suite
- b. Prioritizing and Sequencing
- c. Annotations, Grouping
- d. Parameterization in Suit file for TestNG
- e. Comparison of Junit 5 and TestNG

5. Frameworks

5.1. Data Driven Framework

Data Driven Testing using
- o Excel Files (Using .xls and .xlsx)
- o CSV files
- o Json Files

5.2. Page Factory and Page Object Model

- Importance of page object model
- Implementing Page Object model

5.3.   Helper Utility functions and Classes

Understanding the importance of modularizing the reusable functions and creating a utility classes / functions.

- For creating browser specific drivers
- For scrolling
- Taking screenshots
- Common / Reusable Functions
- Implementing Page Object model
  - Without and with PageFactory concepts

5.4.   Wrapper methods

Understanding the need for wrapper methods and implementing / using wrapper methods

6. Selenium Grid

6.1.   Selenium Grid on local machine

- Learn about the built-in grid functionality in Selenium 3.x
- Learn to distribute the tests on several machines and do parallel execution
- Learn to run the tests in parallel on multiple combinations of browser and OS from a central hub
- Learn to setup the Hub server
- Learn to parameterize the tests to run on various nodes

6.2.   Selenium Grid on cloud machine using Docker

- Learn how to create Selenium Grid on cloud machine
- Understand the basics of Docker containers
- Using docker compose creating Grid
- Learn how to distribute the tests on the remote docker grid

7. Test Reports using Allure and TestNG

7.1.   Using Allure for Reporting

7.2.   TestNg Reports

8. **Running Selenium Tests in Continuous Integration Environment**

    8.1.   **Using Cloud based Jenkins server with Maven plugin**

    8.2.   **Triggering tests on code push to Git repository**

Post CP-SAT Program

It is recommended that mock exams from the CPSAT website are used for practicing. Mock exams for both Theory / Multiple Choice Section and Practical section are available on the website on the following URL

https://cpsat.agiletestingalliance.org/downloads