



## Sample Exam

**Exam Name:** Certified Advanced Selenium Testing Professional (CASTP)

**Exam Code:** CASTP-001

**Question: Which architectural model does Selenium WebDriver primarily use to communicate between test scripts and browsers?**

- A. Peer-to-peer model
- B. Monolithic model
- C. Client-server model
- D. Event-driven model

Correct Answer: C.

---

**Question: What is the primary role of browser-specific drivers such as ChromeDriver and GeckoDriver in Selenium WebDriver architecture?**

- A. To execute test scripts written in different programming languages
- B. To convert WebDriver commands into browser-specific actions
- C. To manage test data and reporting
- D. To provide parallel execution by default

Correct Answer: B

---

**Question: Which optimization technique significantly improves test reliability by synchronizing test execution with dynamic web applications?**

- A. Disabling browser plugins
- B. Using implicit waits only
- C. Implementing explicit, fluent, and custom wait strategies
- D. Increasing test execution timeout globally

Correct Answer: C

**Question: Why is Selenium Grid considered essential for advanced browser and driver management?**

- A. It automatically fixes flaky test scripts
- B. It allows execution of tests only on local machines
- C. It supports distributed and parallel test execution across browsers and environments
- D. It replaces the need for browser drivers

**Correct Answer:** C

---

**Question: Which scenario most strongly justifies the use of XPath over CSS Selectors in Selenium test automation?**

- A. When faster execution speed is the primary requirement
- B. When selecting static elements with unique IDs
- C. When navigating complex relationships such as ancestors, siblings, or descendants
- D. When applying styling rules to web elements

**Correct Answer:** C

---

**Question: What is the most effective Selenium approach to handle AJAX-based elements that load unpredictably without a full page refresh?**

- A. Using Thread.sleep() with a fixed delay
- B. Relying solely on implicit waits
- C. Synchronizing using WebDriverWait with ExpectedConditions
- D. Refreshing the browser until elements appear

**Correct Answer:** C

---

**Question: What key capability distinguishes Fluent Wait from Explicit Wait in Selenium?**

- A. Fluent Wait applies globally to all elements
- B. Fluent Wait allows customization of polling intervals and ignored exceptions
- C. Fluent Wait executes JavaScript asynchronously
- D. Fluent Wait reduces browser memory usage

**Correct Answer:** B

**Question: Why are relative XPath expressions preferred over absolute XPath when dealing with dynamic web applications?**

- A. Relative XPath executes faster in all browsers
- B. Relative XPath is easier to read for beginners
- C. Relative XPath is more resilient to changes in DOM structure
- D. Relative XPath automatically handles AJAX calls

**Correct Answer: C**

---

**Question: Which Selenium strategy best minimizes flaky tests caused by elements that appear or change state dynamically?**

- A. Increasing implicit wait duration globally
- B. Using explicit or fluent waits tied to specific conditions
- C. Hard-coding delays using Thread.sleep()
- D. Refreshing the browser between each test step

**Correct Answer: B**

---

**Question: In which situation is implementing a custom wait strategy most appropriate?**

- A. When waiting for a single element to become visible
- B. When implicit waits are already configured
- C. When multiple complex conditions or asynchronous operations must be validated
- D. When browser-specific drivers are outdated

**Correct Answer: C**

---

**Question: Why is it necessary to switch the WebDriver context when interacting with elements inside an iFrame?**

- A. iFrames load content asynchronously by default
- B. Elements inside an iFrame belong to a different DOM context
- C. iFrames can only be accessed using JavaScript
- D. Selenium does not support locating elements inside iFrames

**Correct Answer: B**