CERTIFICATION SYLLABUS

Certified Appium Automation Specialist

Exam Code: CST-MOB™



A complete preparation guide covering exam objectives, core competencies, domain weightage, and topics to prepare.

©2026 Certaining, Inc. All rights reserved.

Table of Contents

1. **Exam Name & Code** 1
2. **Table of Contents** 2
3. **Objective of the Certification** 3
4. **Core Competencies** 4-6
   - *Mobile Automation Fundamentals & Appium Architecture*
   - *Locators & Element Interaction*
   - *Test Framework Design & Design Patterns*
   - *Test Execution & Device Management*
   - *Cross-Platform & Device Coverage*
   - *CI/CD Integration & DevOps Practices*
   - *Reporting, Logging & Debugging*
   - *Best Practices, Maintenance & Scalability*
5. **Who Should Take This Certification** 7
6. **Exam Information** 7
7. **Examination Domain & Weightage** 8
8. **Topics to Prepare** 9-10
   - *Mobile Automation Fundamentals & Appium Architecture*
   - *Locators & Element Interaction*
   - *Test Framework Design & Design Patterns*
   - *Test Execution & Device Management*
   - *Cross-Platform & Device Coverage*
   - *CI/CD Integration & DevOps Practices*
   - *Reporting, Logging & Debugging*
   - *Best Practices, Maintenance & Scalability*

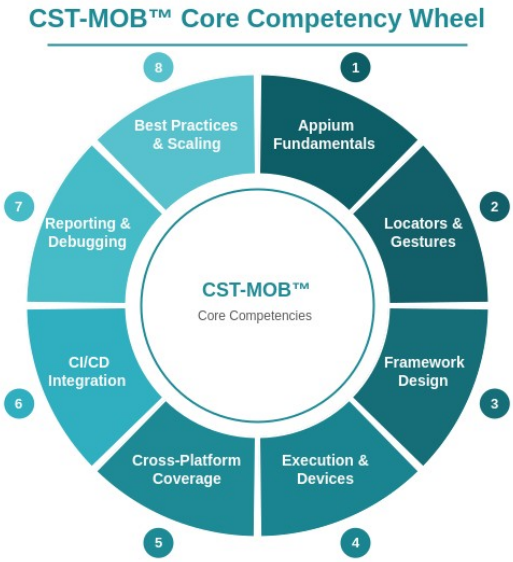
©2026 Certaining, Inc. All rights reserved.

Objective of the Certification

- **Build a strong foundation in Appium architecture —** understanding how Appium uses the WebDriver protocol, the Appium server, and platform drivers (such as UiAutomator2 and XCUITest) to control Android and iOS apps.
- **Develop proficiency in mobile locators and element interaction —** reliably identifying and interacting with elements using accessibility ID, resource ID, XPath, and platform-specific strategies.
- **Establish competency in mobile test framework design —** building maintainable, scalable frameworks using design patterns such as the Page Object Model (POM).
- **Build skills in test execution and device management —** configuring capabilities, managing app installation and state, and running data-driven and parallel tests.
- **Develop cross-platform and device coverage capability —** ensuring automated tests run reliably across Android and iOS, real devices and emulators/simulators, and different screen sizes and OS versions.
- **Introduce CI/CD integration practices —** integrating mobile test suites into pipelines (e.g., Jenkins, GitHub Actions, Azure DevOps) and cloud device farms.
- **Build competency in reporting, logging, and debugging —** generating actionable reports and diagnosing mobile-specific failures efficiently.
- **Establish best practices for stability and scalability —** reducing flaky tests, handling gestures and asynchronous behavior, and keeping suites maintainable as apps evolve.
- **Prepare candidates to contribute effectively to mobile QA teams —** equipping them to build, extend, and maintain real-world mobile automation suites.
- **Create a foundation for progression toward advanced automation and SDET roles —** such as mobile framework architecture, performance testing, or automation leadership.

©2026 Certaining, Inc. All rights reserved.

Core Competencies — CST-MOB™ (Appium Automation Specialist)



©2026 Certaining, Inc. All rights reserved.

1. Mobile Automation Fundamentals & Appium Architecture

- Explain the Appium architecture, including the Appium server, drivers, and the WebDriver protocol.
- Distinguish between native, hybrid, and mobile web apps and how each is automated.
- Set up and configure an Appium environment (server, drivers, SDKs, emulators/simulators, language bindings).

2. Locators & Element Interaction

- Apply locator strategies (accessibility ID, resource ID, XPath, class name, Android UIAutomator, iOS predicate/class chain) and choose the most resilient option.
- Use Appium Inspector to explore the app's element hierarchy and validate locators.
- Perform mobile interactions: taps, text entry, scrolling, swipes, long press, and multi-touch gestures.

3. Test Framework Design & Design Patterns

- Design and implement the Page Object Model (POM) for maintainable mobile test code.
- Structure a framework (folder organization, configuration management, reusable utilities).
- Apply design principles that reduce duplication and support both Android and iOS in one codebase.

4. Test Execution & Device Management

- Configure capabilities for platform, device, app path, and session behavior.
- Manage app lifecycle: install, launch, reset, background, and terminate.
- Run data-driven tests using TestNG/JUnit (or equivalent) and execute tests in parallel across devices.

5. Cross-Platform & Device Coverage

- Execute tests across Android and iOS, and compare real devices with emulators/simulators.
- Handle native and web view contexts in hybrid apps.
- Use cloud device farms (e.g., BrowserStack, Sauce Labs) and manage OS version and screen-size fragmentation.

6. CI/CD Integration & DevOps Practices

- Integrate mobile test suites into CI/CD pipelines (e.g., Jenkins, GitHub Actions, Azure DevOps, GitLab CI).
- Configure automated triggers (on commit, scheduled runs, pre-release gates).

©2026 Certaining, Inc. All rights reserved.

- Understand how mobile automation fits into a broader DevOps/continuous testing strategy.

7. Reporting, Logging & Debugging

- Generate and interpret test reports (e.g., TestNG/JUnit reports, Extent Reports, Allure) with screenshots.
- Capture device logs, Appium server logs, and screen recordings for troubleshooting.
- Debug common failures: locator issues, synchronization problems, permission dialogs, session startup errors.

8. Best Practices, Maintenance & Scalability

- Apply best practices for readable, reusable, and maintainable mobile automation code.
- Reduce flakiness through explicit waits, stable locators, and clean app state between tests.
- Plan for framework evolution as the app UI, OS versions, and device coverage change.

©2026 Certaining, Inc. All rights reserved.

Who Should Take This Certification

- Manual QA testers transitioning into mobile test automation roles.
- Mobile QA engineers responsible for Android and iOS app testing.
- SDETs building or maintaining mobile automation frameworks.
- Mobile developers contributing to test automation as part of Agile/DevOps teams.
- Automation leads/architects evaluating or standardizing team mobile testing practices.
- Individuals who have completed CSTF™ (or hold equivalent foundational testing knowledge) and want to specialize in mobile automation.

Prerequisite: None mandatory. Completion of CSTF™ (Software Testing Foundations) or equivalent practical testing experience is recommended.

Exam Information

Exam Name	Appium Automation Specialist
Exam Code	CST-MOB™
Exam Format	Multiple Choice Questions (MCQ) & Code-Based/Practical Scenario Questions
Duration	120 minutes
Number of Questions	75
Passing Score	70%
Mode of Exam	Test Center Based Testing / Online Testing
Language	English
Validity	Lifetime
Open Book	No

©2026 Certaining, Inc. All rights reserved.

Examination Domain & Weightage

Domain	Weightage
Mobile Automation Fundamentals & Appium Architecture	15%
Locators & Element Interaction	15%
Test Framework Design & Design Patterns	20%
Test Execution & Device Management	15%
Cross-Platform & Device Coverage	10%
CI/CD Integration & DevOps Practices	10%
Reporting, Logging & Debugging	8%
Best Practices, Maintenance & Scalability	7%
Total	**100%**

©2026 Certaining, Inc. All rights reserved.

Topics to Prepare

The topics below represent the core areas candidates should focus on for each domain, reflecting the practical, hands-on depth expected at the CST-MOB™ level.

1. Mobile Automation Fundamentals & Appium Architecture

- **Appium Architecture:** Appium server, platform drivers (UiAutomator2, XCUITest), and the WebDriver protocol.
- **App Types:** Native, hybrid, and mobile web apps, and how automation differs for each.
- **Environment Setup:** Installing Appium, drivers, Android SDK/Xcode, and configuring emulators and simulators.

2. Locators & Element Interaction

- **Locator Types:** Accessibility ID, resource ID, XPath, class name, Android UIAutomator, iOS predicate string and class chain.
- **Appium Inspector:** Inspecting the element hierarchy and testing locators.
- **Gestures:** Tap, swipe, scroll, long press, and multi-touch using W3C Actions.
- **Waits & Synchronization:** Explicit waits to handle dynamic screens and loading states.

3. Test Framework Design & Design Patterns

- **Page Object Model (POM):** Structuring screen classes to separate test logic from locators.
- **Framework Structure:** Organizing folders, configuration files, and reusable utility classes.
- **Cross-Platform Design:** Sharing test logic between Android and iOS while isolating platform-specific locators.
- **Driver Management:** Creating and closing driver sessions cleanly, including factory-style setup.

4. Test Execution & Device Management

- **Capabilities:** Platform name, device name, app path, automation name, and session options.
- **App Lifecycle:** Installing, launching, resetting, backgrounding, and terminating apps.
- **Data-Driven Testing:** Supplying test data via Excel, CSV, JSON, or databases.
- **Parallel Execution:** Running tests concurrently across multiple devices or emulators.

5. Cross-Platform & Device Coverage

- **Android vs. iOS:** Platform differences in locators, permissions, and behavior.
- **Real Devices vs. Emulators/Simulators:** Trade-offs in speed, accuracy, and cost.
- **Hybrid App Contexts:** Switching between native and web view contexts.

©2026 Certaining, Inc. All rights reserved.

- **Device Farms:** Using cloud platforms and managing OS version and screen-size fragmentation.

6. CI/CD Integration & DevOps Practices

- **Pipeline Integration:** Incorporating mobile suites into Jenkins, GitHub Actions, Azure DevOps, or GitLab CI.
- **Automated Triggers:** Running tests on commit, on a schedule, or as a pre-release gate.
- **Continuous Testing Mindset:** Understanding mobile automation's role within a broader DevOps strategy.

7. Reporting, Logging & Debugging

- **Test Reporting:** Generating and interpreting reports (TestNG/JUnit, Extent Reports, Allure).
- **Logs & Evidence:** Capturing Appium server logs, device logs, screenshots, and screen recordings.
- **Debugging Failures:** Diagnosing locator issues, timing problems, permission pop-ups, and session startup errors.

8. Best Practices, Maintenance & Scalability

- **Code Quality:** Writing readable, reusable, and maintainable automation code.
- **Flakiness Reduction:** Stable locators, explicit waits, and resetting app state between tests.
- **Framework Evolution:** Planning for growth as the app UI, OS versions, and device coverage change.

©2026 Certaining, Inc. All rights reserved.