

Android Testing

Course Summary

Description

Android Testing is a three-day training course focusing specifically on tools and techniques to effectively integrate automated testing into your application development workflow.

Through lecture and hands-on exercises, you will learn the basics of how to build and organize a robust automated test suite for the Android platform. In this training course we will explore how the latest Android Testing Support Library offerings from Google provides developers with a platform that greatly simplifies test automation—and we will compare it to other popular 3rd party solutions.

We also focus on how to integrate the automated tests you've written into continuous integration (CI) and continuous delivery (CD) pipelines, to produce well-tested code at any scale!

Objectives

By the end of this course, students will be able to:

- Explain the components of your mobile application that should be tested
- Read and analyze test reports and metrics from both Android Studio and Gradle
- Discern how local JVM unit tests can improve your development workflow
- Build and execute instrumented unit tests in Android Studio
- Construct a test suite that is useful and maintainable over the long haul
- Leverage test automation in continuous integration (CI) and continuous delivery (CD) environments

Topics

- Designing Automated Tests
- Testing in Android Studio
- Writing Effective Unit Tests
- Writing Effective Instrumentation Tests
- Building Sustainable Tests
- Scaling Test Automation

Audience

This course is designed for experienced Android developers and QA professionals looking to improve the quality of their application code through test automation.

Prerequisites

In order to get the most out of this training, students should:

- Have taken Android Overview training or any other NewCircle Android class that contains the Android Overview module.
- It is highly recommended that participants be familiar with basics of Java, C/C++, and Linux administration to fully take advantage of this course.
- To refresh your Java skills, you can review NewCircle's Fundamentals of Java tutorial
- Additionally, "bootcamp-level" knowledge of Android Studio and the Gradle build system is required

Duration

Three days

Android Testing

Course Outline

I. Designing Automated Tests

This introductory module provides some high level context around the basic design of test cases for mobile applications. By the end of this module, you will be able to answer the questions "what should I be testing?" and "what type of tests should I have?"

- A. Students can feel that they successfully completed this section if they:
- B. Deterministic vs. non-deterministic tests
- C. Hermetic test environments
- D. Organizing your test suite
- E. Mobile testing best practices
- F. Android testing best practices

II. Testing in Android Studio

The Android Testing Support Library has made great strides in integrating automated test with the build and development tools. In this module, you will learn to integrate automated tests into your project, and how to execute them from Android Studio and Gradle command line builds.

- A. Junit 4 syntax and style
- B. The AndroidJUnitRunner
- C. Difference between JVM and instrumented unit tests
- D. Methods of executing automated test suites

III. Writing Effective Unit Tests

What makes a good unit test? Unit tests should be a reliable safety net when refactoring existing code or implementing a new feature—but how do we achieve this? In this module, we will explore techniques for creating tests that meet the design objectives set forth in previous modules. You will also gain experience writing and running unit tests on the local JVM.

- A. Treating tests as functional documentation
- B. Readable tests through fluent assertions
- C. Interpreting code coverage reports
- D. Mocking external dependencies
- E. Writing tests that expect failure

IV. Writing Effective Instrumentation Tests

In many cases, local JVM unit tests aren't enough—at some point you need to get your code onto real devices. This module will teach you how the Android instrumentation framework works. You will learn how Google's instrumentation test frameworks enable writing tests that are easy to write, easy to read, and return repeatable results. You will also gain experience writing and running instrumentation tests.

- A. Android instrumentation overview
- B. Android-specific JUnit rules
- C. Espresso overview
- D. UI Automator overview
- E. Espresso usage and optional packages
- F. Building UI Automator tests
- G. Synchronization with UI Automator and Espresso

V. Building Sustainable Tests

Good tests don't require much maintenance. Your automated test code should be something you come to rely on, rather than something you spend more time maintaining than the application itself. This module will show you several techniques you can use to make your test suite maintainable, configurable, debuggable, and compatible.

- A. Page object models
- B. Parameterized testing with AndroidJUnitRunner
- C. Handling flaky test cases
- D. JUnit RunListener
- E. Abstracting system interactions
- F. Test filtering with Android JUnitRunner

VI. Scaling Test Automation

At some point, running build and test procedures on your local development machine isn't going to cut it. In this module, you will learn how Google's test frameworks make integrating with continuous integration (CI) and continuous delivery (CD) workflows seamless. We will explore metrics to consider when deciding how to test your application at scale. You will also gain experience pushing your automated test suite through the CI process.

- A. Role of automated build servers
- B. Source code management techniques
- C. Selecting the devices in your test pool
- D. In-house test labs vs. cloud device farms
- E. Creating useful build result dashboards
- F. Instrumenting delivery of successful builds