INDEX

No.	Definition / Aim	CO	Date	Signature
13	Create a Python-Appium automation script to interact with checkboxes, radio buttons, switches, and other selectable mobile UI controls and verify their states.	CO3		
14	Automate an Android application using ID, Accessibility ID, Class Name, XPath, and other suitable locators and compare their practical usage.	CO3		
15	Develop a robust Python-Appium test by handling waits, element-not-found errors, session errors, application launch problems, and other common automation issues.	CO3		
16	Capture an Android application screen using UiAutomatorViewer and identify UI objects, resource IDs, text, class names, package names, and element bounds.	CO4		
17	Configure ADB and execute practical commands to identify devices, install/uninstall APKs, launch applications, capture logs, and perform basic device operations.	CO4		
18	Enable USB debugging and connect a real Android device to the computer using ADB, then verify the device and perform basic testing operations.	CO4		
19	Develop a Python-based automation workflow that uses ADB and Appium to prepare the device, launch an application, execute test steps, and collect test results.	CO4		
20	Develop and execute a complete Python-Appium test suite covering application launch, element identification, user input, navigation, validation, error handling, and test result reporting.	CO4		

Pra-01	Install and configure Python, Android Studio, Android SDK, Android Emulator, Appium Server, Appium Inspector and required Python packages for mobile automation.
Code Files	
Input	
Output	
Screenshot	