**Class 1: TestRunner | TestNG Usage | Browser Factory**

1. TestNG Installation
2. TestNG @Test Annotation: To identify test as TestNG Test. Ex TC\_SP\_011
3. TestNG @Parameters: Pass Browser type from TestNg XML file to Test. Ex TC\_SP\_011\_1
4. Browser Factory: Class responsible for Driver initialization of different type. Ex TC\_SP\_011\_2: **Factory Pattern or Singleton pattern**
5. TestNG Asserts: Apply verifications to the test. Ex TC\_SP\_011\_3

**Practice Exercise**

1. Convert all the Sanity Pack test to TestNG Test
2. Use Browser Factory in all Tests to Initialize Browser and Quit Browser
3. Convert all the Verification in the tests to TestNG Asserts

**Class 2: Page Object Model | Selenium Page Factory**

1. Page Object Model with Public members and implementing Selenium Page Factory. Ex LoginPage\_0 & TC\_SP\_11\_PageObjectModel\_0
2. Convert Public members to Private and expose the members with getter setter. Ex LoginPage\_1 & TC\_SP\_11\_PrivateMember\_1
3. Combine multiple sequences of Actions in to one method. Ex LoginPage\_2 & TC\_SP\_11\_CombineActions\_2
4. Create a BasePageObject class for all the common functionality which can be extended by all the PageObject classes. Ex LoginPage\_2 & TC\_SP\_11\_BasePageObject\_3

**Practice Exercise**

1. Write PageObjects for all the Pages used across the Tests

**Class 3: Complex Page Object Model | Reading Configurations**

1. Selenium Page Factory @FindsAll Annotation: To get collections of same type of elements. Ex TC\_SP\_005\_1 & CartPage\_1
2. Complex Page Objects with in Container: Initialize object with in PageObjects. Ex TC\_SP\_005\_2 & CartPage\_2
3. Reading Configurations Property Files: Read Browser type from Config file. Ex TC\_SP\_005\_3

**Practice Exercise**

1. Put Environment url & ImplicitlyWait in to config file
2. Create CategoryPage & CategoryItem using same container technique and complete TC\_SP\_003 & TC\_SP\_004
3. Create WishListItem using same container technique and complete TC\_SP\_007
4. Convert all TestScripts to latest Page Object Model

**Class 4: Factory Design Pattern | Base Test Class | Excel Reader**

1. ConfigReaderFactory: Implement Factory Design Pattern to read all the different config properties. Ex TC\_SP\_011\_1
2. BaseTestClass: Implement Abstract Base Class for sharing common members with all the Tests and to set up Before and After method. Ex TC\_SP\_011\_2 & BaseTestClass
3. ExcelReader: Set up ExcelTestDataReader to use ApachePOI for reading Data from Excel Test Data Sheet. Ex TC\_SP\_011\_3

**Practice Exercise**

1. Extend BaseTest class for all the Sanity Pack test Cases
2. Remove Config and Driver Instantiation from the test cases, as the same has been done in the beforeTest method of BaseTest class.
3. Remove Driver.Quit from the entire tests, as same has been implemented in the afterTest method of BaseTest class.

**Class 5 Extent Reporter | TestNg Listeners**

1. Extent Report Installation.

SetUp ExtentReport Class: Involve before and after suite method in the BaseTestClass to start and close the reporting. Logging within the test. Ex TC\_SP\_001\_1

1. Implement ITestListeners to catch the Assert Failure and to end the test logically. Make use of onTestSuccess and onTestFailure. Ex TC\_SP\_001\_2. Add @Listeners(listeners.TestListener.class) to BaseTestClass.
2. Make reporter class available to PageObjects as well by including it in the PageObjectBase class.
3. Move TestName to baseClass.BeforeMethod and create ExcelConsts to avoid spelling mistakes Ex: TC\_SP\_011\_4

**Practice Exercise**

1. Take all the HardCoded values from the ExcelTestData sheet, for Ex Product Name, Colour Size. Make sure to enter column Names to ExcelConsts class.
2. Write report logs for the entire tests or inside all page objects.

**Class 6: Smart/Customer Exception Handling**

1. How to Throw Custom Exceptions. Ex Move the ImplicitlyWait & Maximise window command from BaseTestClass to WebDriverFactory class and handle the exception.
2. Throw Exception for incorrect BrowserName value in WebDriverFactory
3. Capture Screenshot on Error: Fit this on onTestFailure method in TestListeners
4. Complete TestScenario\_3

**Practice Exercise**