----

1. **Learnings from Course (UDEMY - OC - Selenium)**
   a. Atomic test cases
      i. UI + API
      ii. Selenium WebDriver + RestAssured
   b. OOPS in Automation framework
      i. Abstraction
         1. WebDriver driver = new ChromeDriver();
      ii. Encapsulation
         1. Page Object Model
            a.
            ```java
            private final By firstNameFld = By.id("billing_first_name");
            private final By lastNameFld = By.id("billing_last_name");
            private final By addressLineOneFld = By.id("billing_address_1");
            private final By billingCityFld = By.id("billing_city");

            public CheckoutPage enterFirstName(String firstName) {

                clearAndSendKeys(firstNameFld, firstName, WaitStrategy.PRESENCE, "First Name Field");
                return this;
            ```
            b. `}`
      iii. Inheritance (is-a)
         1. TestScript extends TestBase
            a. @BeforeMethod - startDriver();
            b. @AfterMethod - quitDriver();
      iv. Composition (has-a)
         1. Many elements are common in some pages
      v. Polymorphism
   c. **Maintain different config.properties files for different environments**
      i. stg_config.properties
      ii. int_config.properties
   d. How to parse HTML response using
      i. Groovy GPath
      ii. JSoup
   e. **Convert RestAssuredCookie to SeleniumCookie**
      i. And, inject Selenium Cookie into Browser
   f. Fluent Interface
   g. Design pattern -
      i. Builder pattern (Chain the methods)
      ii. Singleton (To load Properties file only once)
      iii. Factory (Driver start)
   h. Page Objects:
      i. Structural (2 methods -> sendKeys() and click())
      ii. Functional (1 method -> sendKeys() and click())
   i. **Browser value**
      i. Run test case individually
      ii. Run test case using testng.xml
      iii. Run test case using maven command line

1. mvn clean test -Dbrowser=CHROME
2. mvn clean test -Dbrowser=EDGE

j. **TestNG file selection using maven:**
   i. mvn clean test -DsuiteFile=testng_smoke.xml
   ii. mvn clean test -DsuiteFile=testng_sanity.xml

k. **Maven command:**
   i. mvn clean test -Dbrowser=CHROME -DsuiteFile=testng_Jenkins.xml
      -DbrowserInstances=4 -Denv=STAGE
   ii. mvn clean test -Dbrowser=FIREFOX -DsuiteFile=testng_Jenkins.xml
      -DbrowserInstances=5 -Denv=STAGE
   iii. mvn clean test -Dbrowser=EDGE -DsuiteFile=testng_Jenkins.xml
      -DbrowserInstances=5 -Denv=STAGE

l. DataDriven testing using JSON file
   i. Serialization (Java object -> JSON)
   ii. De-Serialization (JSON -> Java object)

```java
/* Generic method -. can be used for any class */
public static <T> T deSerializationJSON(String fileName, Class<T> T) throws IOException {
    /* myBillingAddress.json --> has to be present under src/test/resources */
    InputStream is = JacksonUtils.class.getClassLoader().getResourceAsStream(fileName);
    ObjectMapper objectMapper = new ObjectMapper();
    return objectMapper.readValue(is, T);
}
```
   iii.

m. Allure reports
n. Github Hooks + Jenkins