- Why we require Testing?

i. To deliver quality product to customer

ii. To test whether software is developed as per customer requirement or not

iii. To check whether software is working as per customer requirement or not

- What are Types of software Testing?

iv. Black Box Testing

v. White Box Testing

vi. Grey Box Testing

- What are Types of Black Box Testing?

vii. Manual Testing

viii. Automation Testing

- What is manual Testing?

ix. In manual testing we simply create test cases and execute test cases and perform certain steps to check whether software is working as per expectations or not

x. In manual testing we check expected result and actual result is matching or not. If matching then test case is passed

- What is automation Testing?

xi. Process of converting manual test cases into the test scripts with the help of automation tools or any programming language

- What are the challenges we faced in manual testing?

xii. Retesting–Execute same test case again and again with multiple sets of data
Ex: We have login screen and we have to enter username and password
We are entering valid username and password is not enough
Here we need to enter positive well as negative scenarios(ex: valid username, invalid password or invalid username, valid password or valid username and no password)
In real time project we have lots fields with lots of data. To enter data into this field will require lot of time and lots of efforts/ also might be chances that we may miss some inputs or tester get tired

xiii. Regression testing–Revalidating defects in newer build and because of this fixes make sure there is no side effects. So for every cycle we check whatever defects raised are fine and

- When we can go for Automation Testing?

xiv. When the cost makes sense

xv. When using repetitive tests

xvi. When time will be saved

xvii. When quality is sure to be improved

xviii. When tests are run frequently

xix. When you need to run multiple tests at once

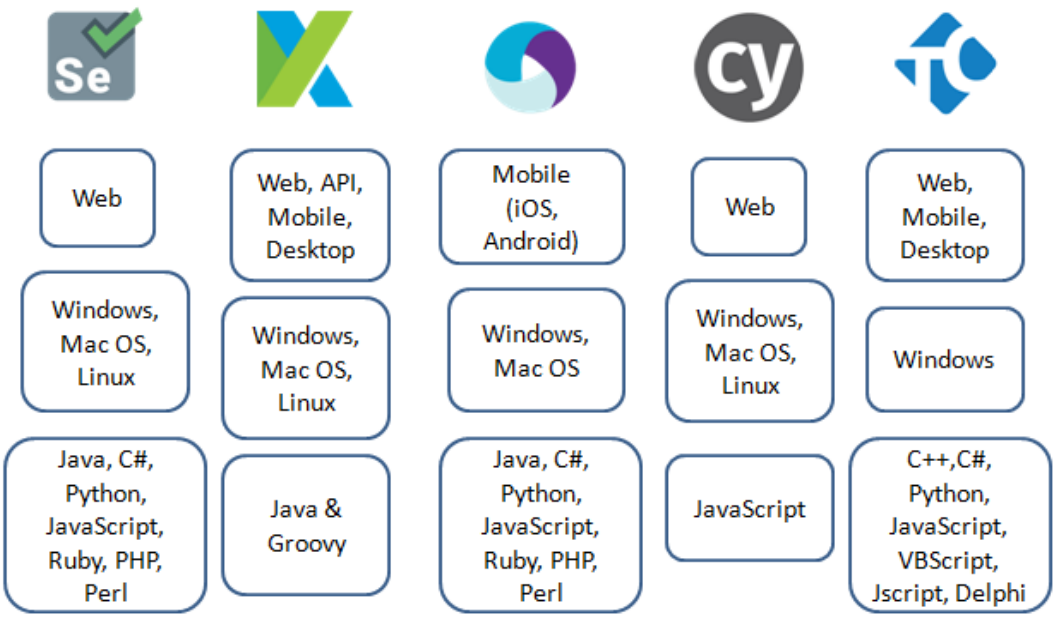
- Manual Vs Automation

• Test cases are executed manually.	• Test cases are executed automatically with the help of tools.
• Difficult to ensure sufficient test coverage	• Easy to ensure greater test coverage.
• May difficult to test on different browsers	• We can easily test on different
• you need to sit in front of your system and execute test cases	• You just have to run Automation scripts you can run it overnight!

- Limitations of Automation Testing

i. When Cannot perform testing for Images.
ii. Captcha, Barcodes.
iii. Continues maintenance of code.
iv. Cannot perform testing for audio or video

TOOLS FOR AUTOMATION TESTING



- SELENIUM & ITS FEATUES

Selenium is a free automated testing tool

i. Open Source
ii. Supports Multiple operating system
iii. Supports multiple browsers
iv. Supports multiple programming languages
v. Supports multiple framework
vi. Supports parallel and cross browser execution

- How to select right tool for Automation?

i. Project Requirements

ii. Team skills / Learning Curve

iii. Budget

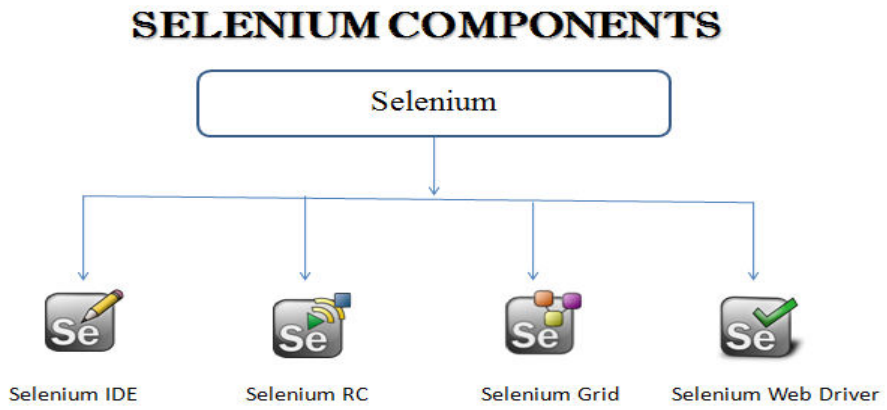
iv. Ease of Test case Creation and Maintenance

v. Reusability

vi. Data-Driven Testing

vii. Reporting

viii. Support for Collaboration



- Prerequisite for selenium

i. Selenium

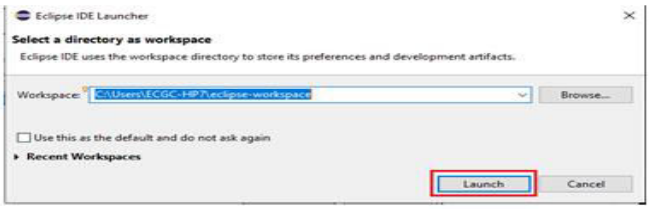
ii. Eclipse

iii. Java

- Installation steps

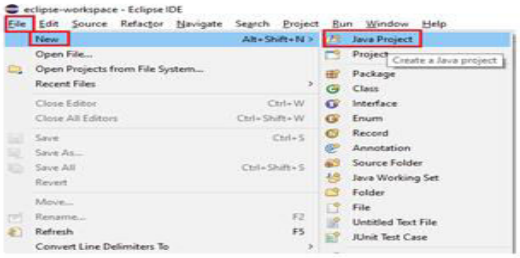
iv.

v.



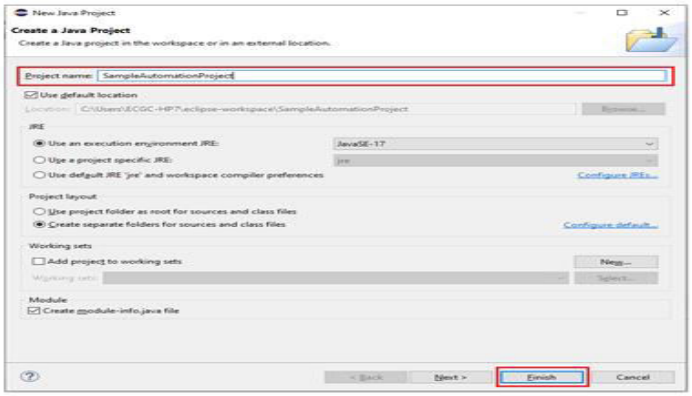
1. Launch **Eclipse** and select the default workspace. Click on **Launch**

2. Click on **File -> New -> Java Project**



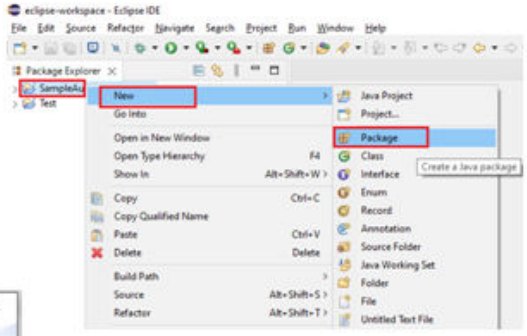
vi.

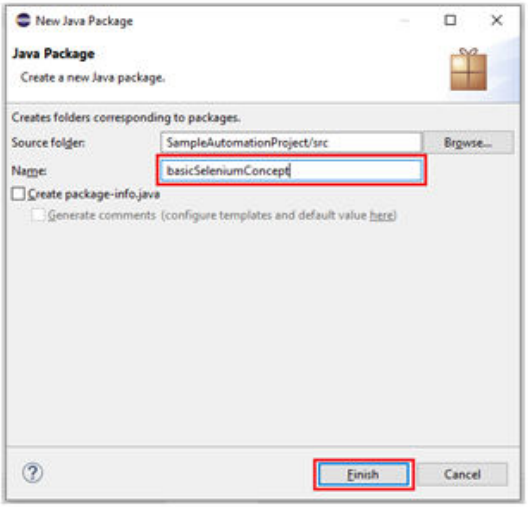
3. Enter a **Project name** and click on **Finish.**



vii.

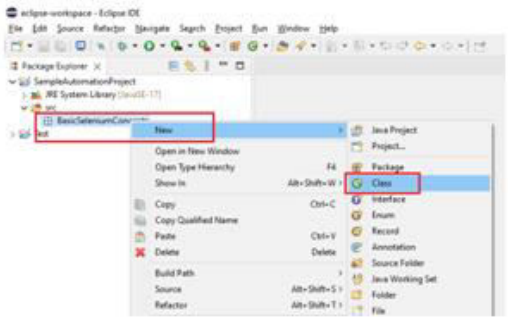
4. **Right Click** on the newly created Project, then **New -> Package**





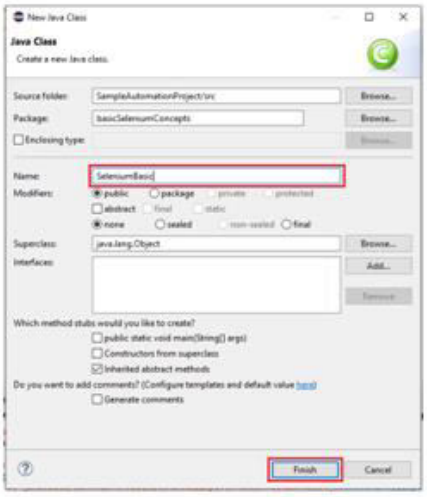
5. Enter **Package Name** and click on **Finish.**

viii.



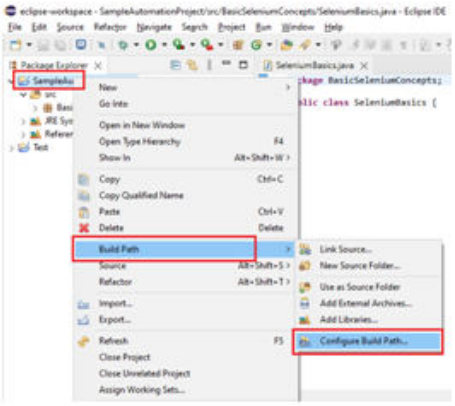
6. **Right Click** on newly created package and then **New -> Class**

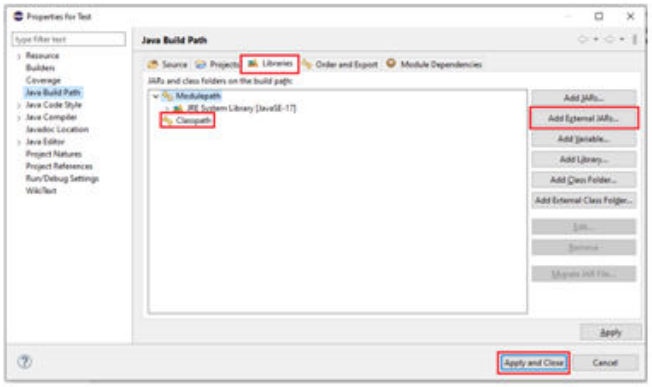
7. Enter **Class Name** and then click on **Finish.**



ix.

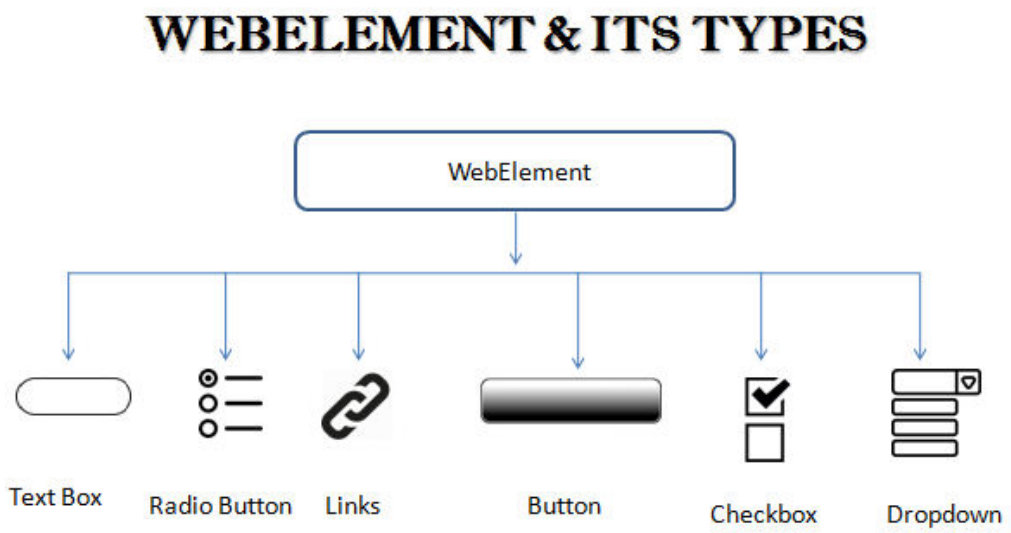
8. **Right Click** on project folder and then **Build Path -> Configure Build Path**





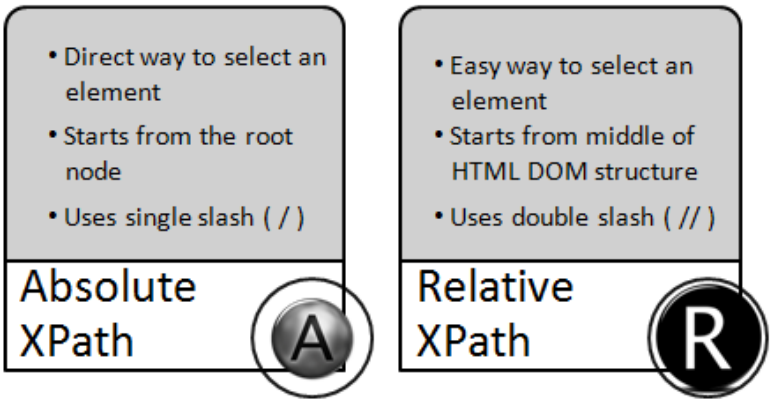
9. Click on **Libraries** and then Select **Classpath**, then click On **Add External JARs.** Add the downloaded Selenium Jar files and then **Apply and Close.**

x.



xi.

XPATH



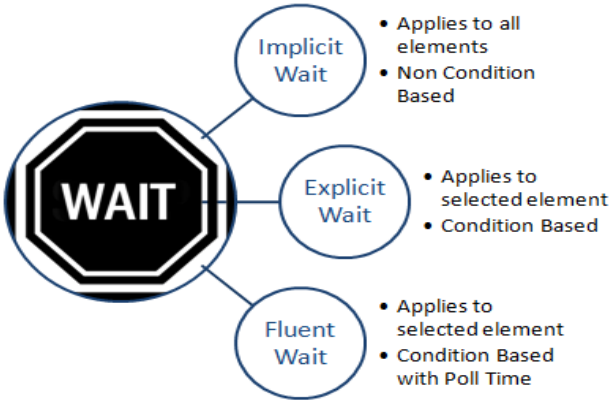
xii.

RELATIVE XPATH

- Basic Xpath
- Contains
- OR & AND
- Starts-with
- Text()
- Last()
- Position()
- Following
- Preceding

xiii.

WAITS



Implicit Wait:

Syntax : driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

Explicit Wait:

Syntax :WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("xpath")));

Methods available:

alertIsPresent()

elementSelectionStateToBe()

elementToBeClickable()

elementToBeSelected()

frameToBeAvaliableAndSwitchToIt()

invisibilityOfTheElementLocated()

invisibilityOfElementWithText()

presenceOfAllElementsLocatedBy()

presenceOfElementLocated()

textToBePresentInElement()

textToBePresentInElementLocated()

textToBePresentInElementValue()

titleIs()

titleContains()

visibilityOf()

visibilityOfAllElements()

visibilityOfAllElementsLocatedBy()

visibilityOfElementLocated()

Fluent Wait:

Syntax:Wait wait = new FluentWait(WebDriver reference)

.withTimeout(timeout, SECONDS)

.pollingEvery(timeout, SECONDS)

.ignoring(Exception.class);

Example:

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)

.withTimeout(30, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);