STQA Mini Project No. 1

1.1 Title

Mini-Project 1: Create a small application by selecting relevant system environment/ platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.

1.2 Problem Definition:

Perform Desktop Application testing using Automation Tool like JUnit generate Test Report by Using tool like Apache Maven.

1.3 Prerequisite:

Knowledge of Core Java, Basic Concepts of Unit Testing, Test Cases Writing using Junit etc tool

1.4 Software Requirements:

JDK 1.8, Eclipse java photon-R version, TestNG

1.5 Hardware Requirement:

PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

1.6 Learning Objectives:

We are going to learn how to Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. also Prepare Test Reports based on Test Pass/Fail Criteria

1.7 Outcomes:

You are able to understand Unit and Integration testing with Tool with Test Report.

1.8 Theory Concepts:

1.8.1 What is Unit Testing?

Unit Testing of software applications is done during the development (coding) of an application.

The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure

The goal of Unit Testing is to isolate each part of the program and show that the individual parts are correct. Unit Testing is usually performed by the developer.

1.8.2 Unit Testing Tools

There are several automated tools available to assist with unit testing. We will provide a few examples below:

1. Jtest: Parasoft Jtest is an IDE plugin that leverages open-source frameworks (Junit, Mockito, PowerMock, and Spring) with guided and easy one-click actions for creating, scaling, and maintaining unit tests. By automating these time-consuming aspects of unit testing, it frees the

developer to focus on business logic and create more meaningful test suites.

2. Junit: Junit is a free to use testing tool used for Java programming language. It provides assertions to identify test method. This tool test data first and then inserted in the piece of code.
3. NUnit: NUnit is widely used unit-testing framework use for all .net languages. It is open source tool which allows writing scripts manually. It supports data-driven tests which can run in parallel.
4. JMockit: JMockit is open source Unit testing tool. It is code coverage tool with line and path metrics. It allows mocking API with recording and verification syntax. This tool offers Line coverage, Path Coverage, and Data Coverage.
5. EMMA: EMMA is an open-source toolkit for analyzing and reporting code written in Java language. Emma support coverage types like method, line, basic block. It is Java-based so it is without external library dependencies and can access to the source code.
6. PHPUnit: PHPUnit is a unit testing tool for PHP programmer. It takes small portions of code which is called units and test each of them separately. The tool also allows developers to use pre-define assertion methods to assert that system behave in a certain manner.

Those are just a few of the available unit testing tools. There are lots more, especially for C languages and Java, but you are sure to find a unit testing tool for your programming needs regardless of the language you use.

1.8.3 Extreme Programming & Unit Testing

Unit testing in Extreme Programming involves the extensive use of testing frameworks. A unit test framework is used in order to create automated unit tests. Unit testing frameworks are not unique to extreme programming, but they are essential to it. Below we look at some of what extreme programming brings to the world of unit testing:

- Tests are written before the code
- Rely heavily on testing frameworks
- All classes in the applications are tested
- Quick and easy integration is made possible

1.8.4 Bug taxonomy

Bug taxonomies help in providing fast and effective feedback so that they can easily identify possible reasons for failure of the software. Using bug taxonomy, a large number of potential bugs can be grouped into few categories.

Whenever a new bug is reported, using bug taxonomy, a tester can easily analyse and put that bug into any of these categories.

At the end of testing, Testers can understand the type of categories of bugs that frequently occurred and thereby in successive rounds of testing he can focus on writing more test cases that would help to detect such bugs. In addition, test leaders can guide their testers to focus on such frequently occurring bugs.

The summary of the Bug Taxonomy is given below,

- Requirements, Features, and Functionality Bugs
- Structural Bugs
- Data Bugs
- Coding Bugs
- Interface, Integration, and System Bugs
- Test and Test Design Bugs
- Testing and Design Style

1.8.5 What is Integration Testing?

In integration Testing, individual software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing

Integration Test Case:

Integration Test Case differs from other test cases in the sense it **focuses mainly on the interfaces & flow of data/information between the modules**. Here priority is to be given for the **integrating links** rather than the unit functions which are already tested.

Sample Integration Test Cases for the following scenario: Application has 3 modules say 'Login Page', 'Mail box' and 'Delete mails' and each of them are integrated logically.

Here do not concentrate much on the Login Page testing as it's already been done in Unit Testing. But check how it's linked to the Mail Box Page.

Similarly Mail Box: Check its integration to the Delete Mails Module.

Test Case ID	Test Case Objective	Test Case Description	Expected Result
1	Check the interface link between the Login and Mailbox module	Enter login credentials and click on the Login button	To be directed to the Mail Box
2	Check the interface link between the Mailbox and Delete Mails Module	From Mail box select the an email and click delete button	Selected email should appear in the Deleted/Trash folder

1.8.6 Desktop Application Testing by Using Junit Tool

What is Junit?

JUnit is a framework for implementing testing in Java.

It provides a simple way to explicitly test specific areas of a Java program, it is extensible and can be employed to test a hierarchy of program code either singularly or as multiple units.

Why use a testing framework? Using a testing framework is beneficial because it forces you to explicitly declare the expected results of specific program execution routes. When debugging it is possible to write a

test which expresses the result you are trying to achieve and then debug until the test comes out positive. By having a set of tests that test all the core components of the project it is possible to modify specific areas of the project and immediately see the effect the modifications have on the other areas by the results of the test, hence, side-effects can be quickly realized.

JUnit promotes the idea of first testing then coding, in that it is possible to setup test data for a unit which defines what the expected output is and then code until the tests pass. It is believed by some that this practice of "test a little, code a little, test a little, code a little..." increases programmer productivity and stability of program code whilst reducing programmer stress and the time spent debugging.

JUnit is a simple open source Java testing framework used to write and run repeatable automated tests.

It is an instance of the xUnit architecture for unit testing framework. Eclipse supports creating test cases and running test suites, so it is easy to use for your Java applications.

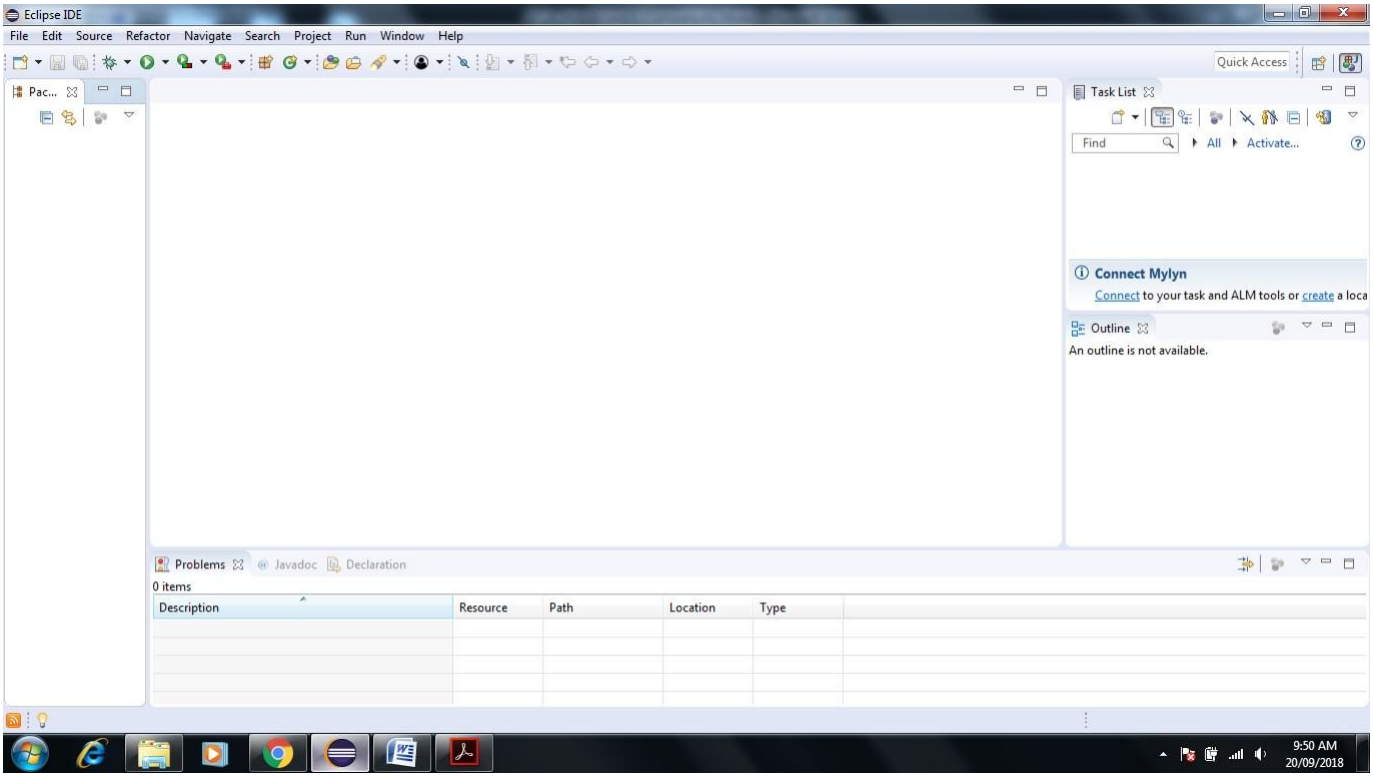
JUnit features include:

- Assertions for testing expected results
- Test fixtures for sharing common test data
- Test suites for easily organizing and running tests
- Graphical and textual test runners

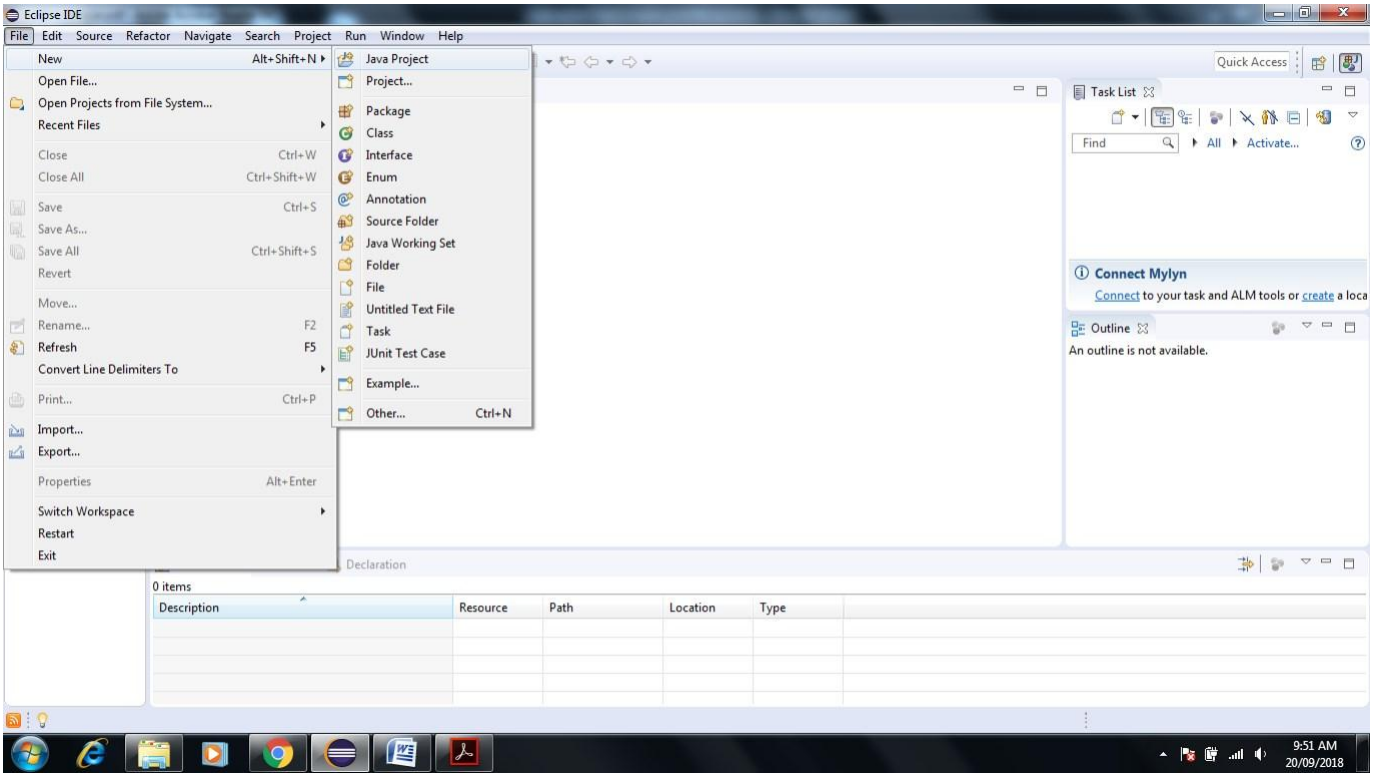
1.8.7 How to Create Simple Junit Test in Eclipse IDE

1. Download JDK 1.8 and Eclipse latest version here we are using eclipse-java-photon-R-win32.

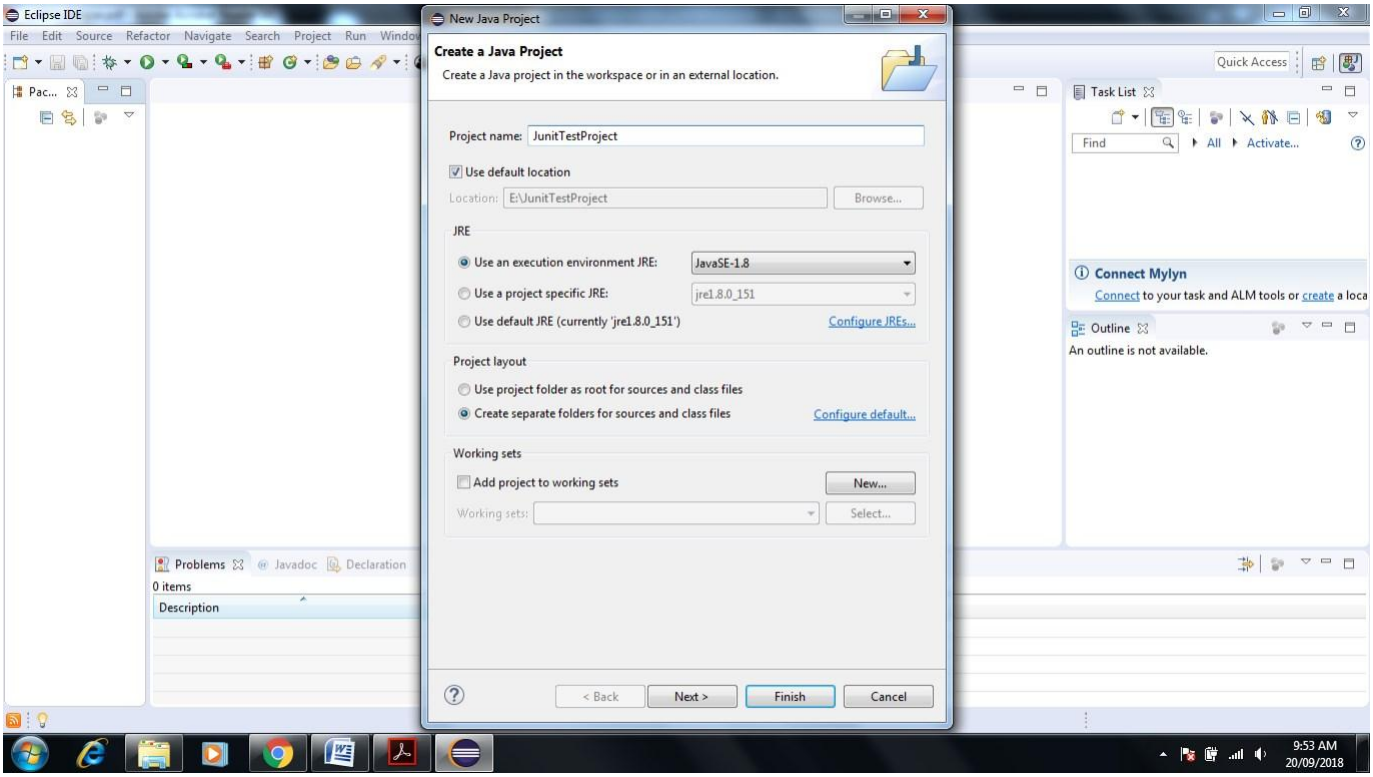
2. Open Eclipse IDE



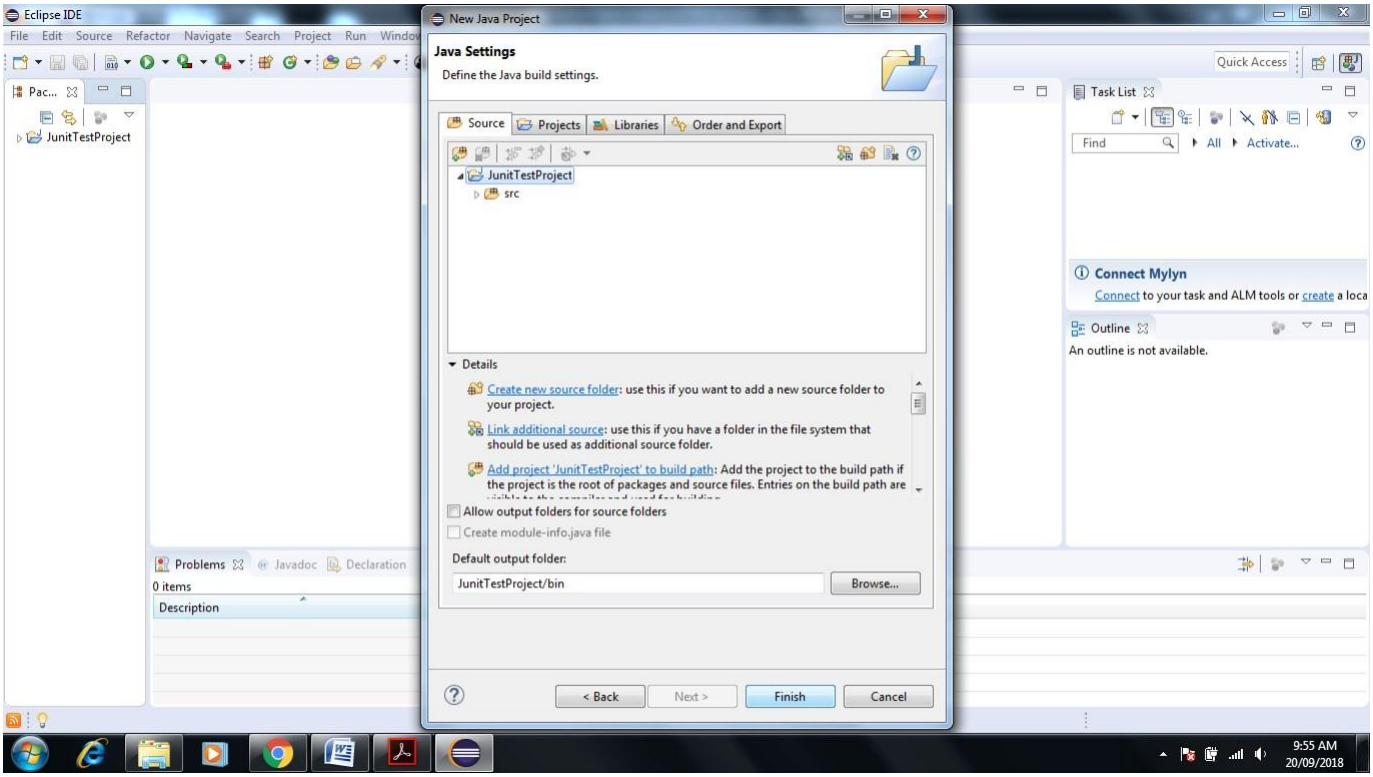
3. Go to File and Select New -> Create New Java Project



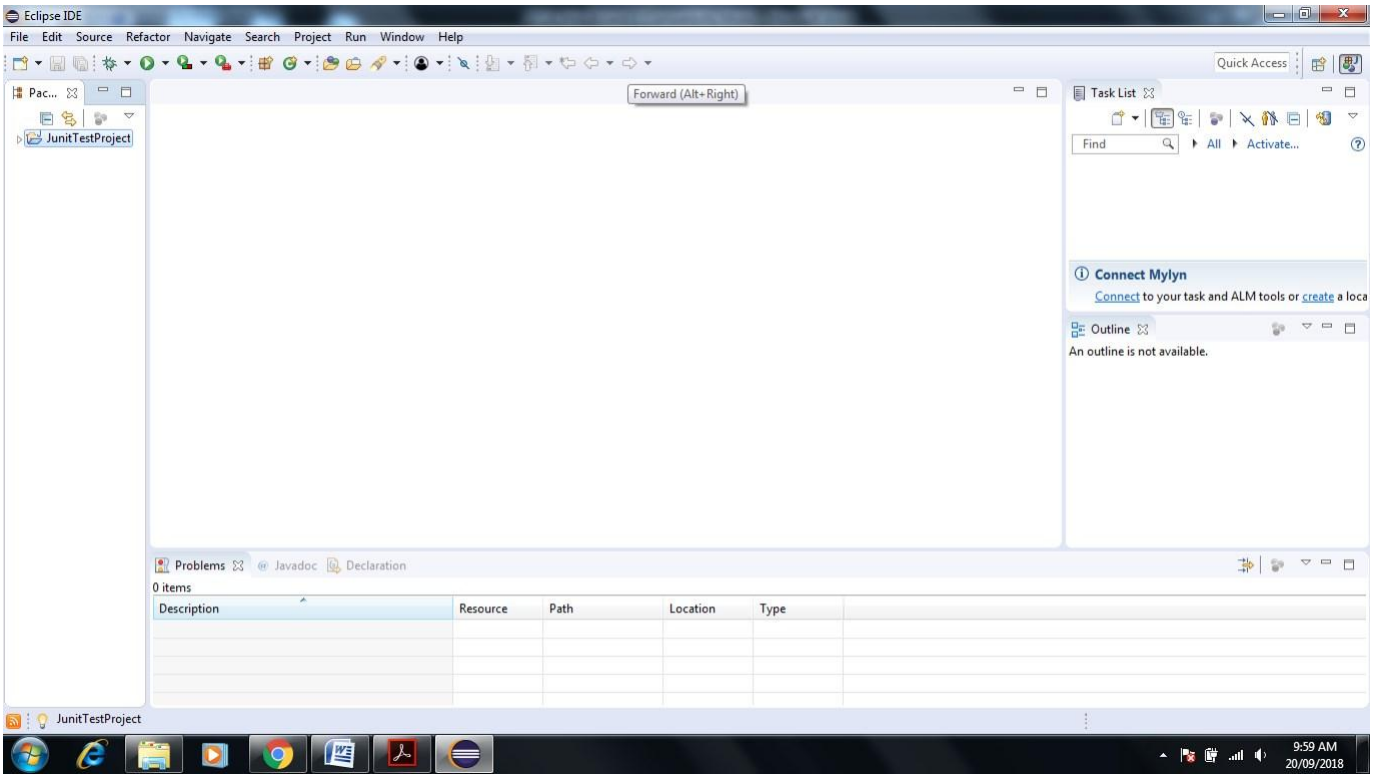
4. Give JunitTestProject name to the project and check use project folder as root for source and class files



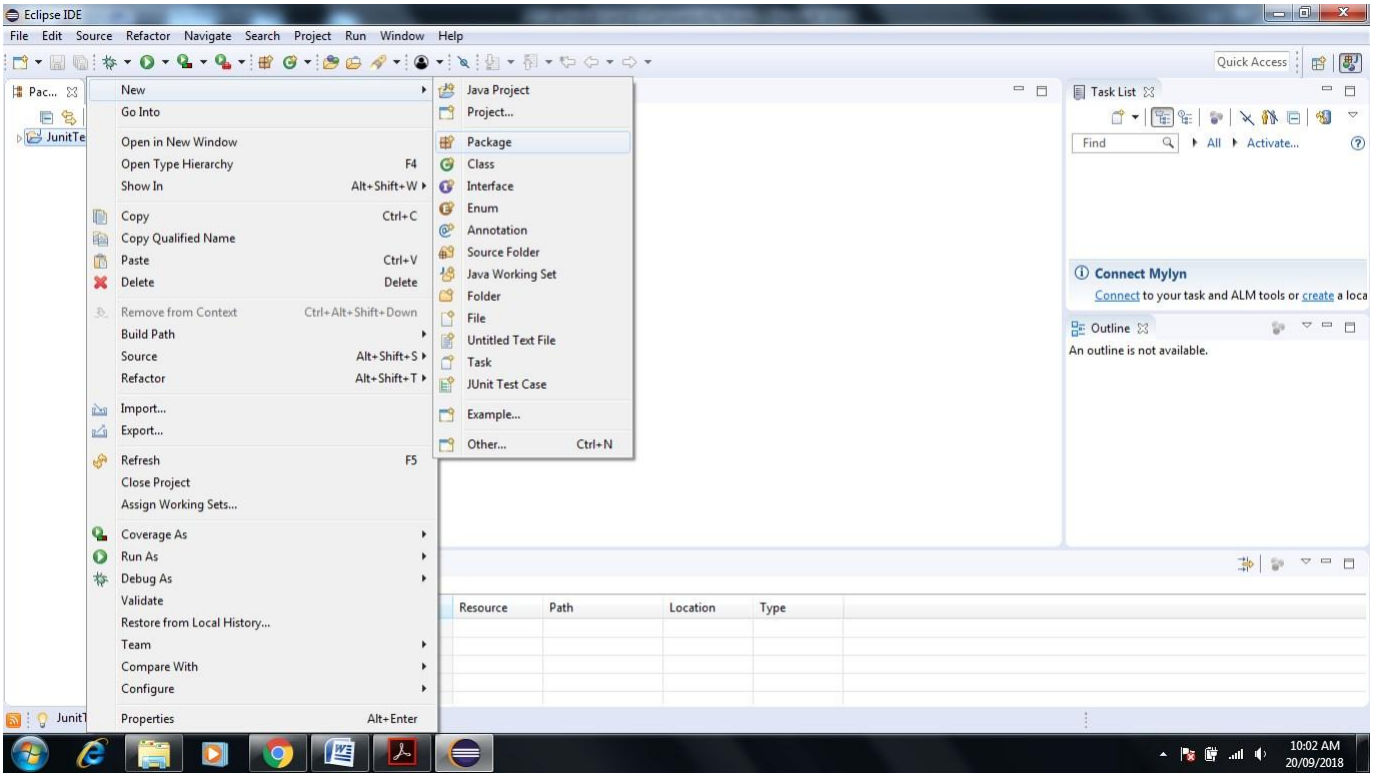
5. Click on Next-> Next Screen will Appear-> Click Finish



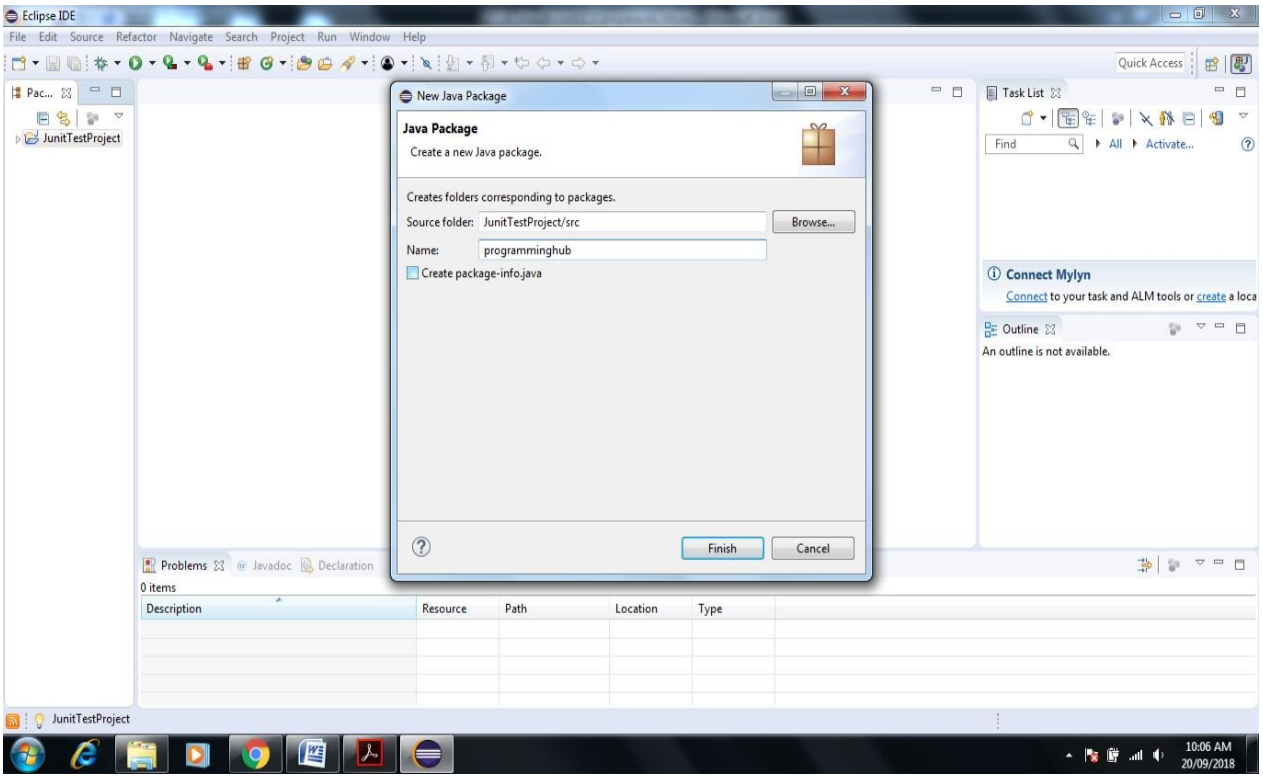
6. Next Screen Shown JunitTestProject Folder in Project Explorer



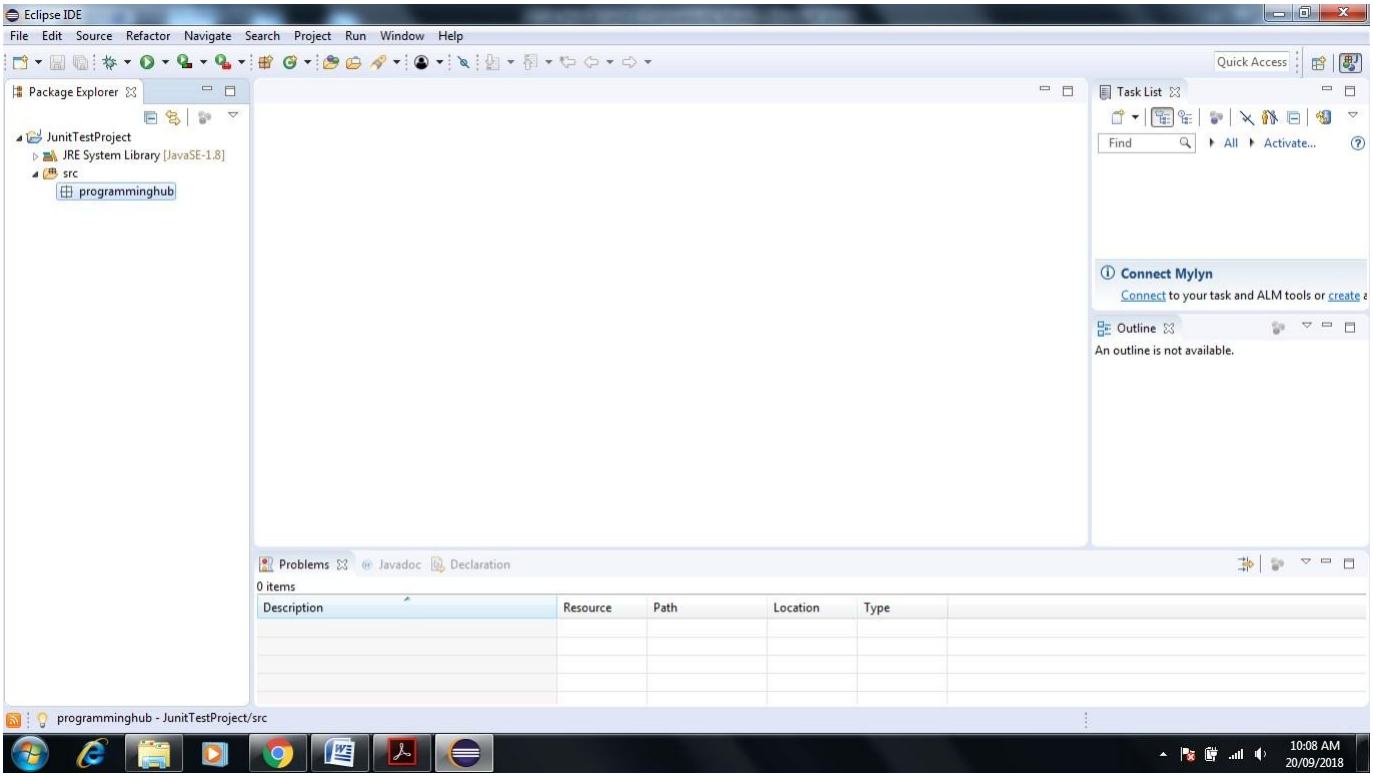
7. Right Click on Folder name JunitTestProject->New->Package



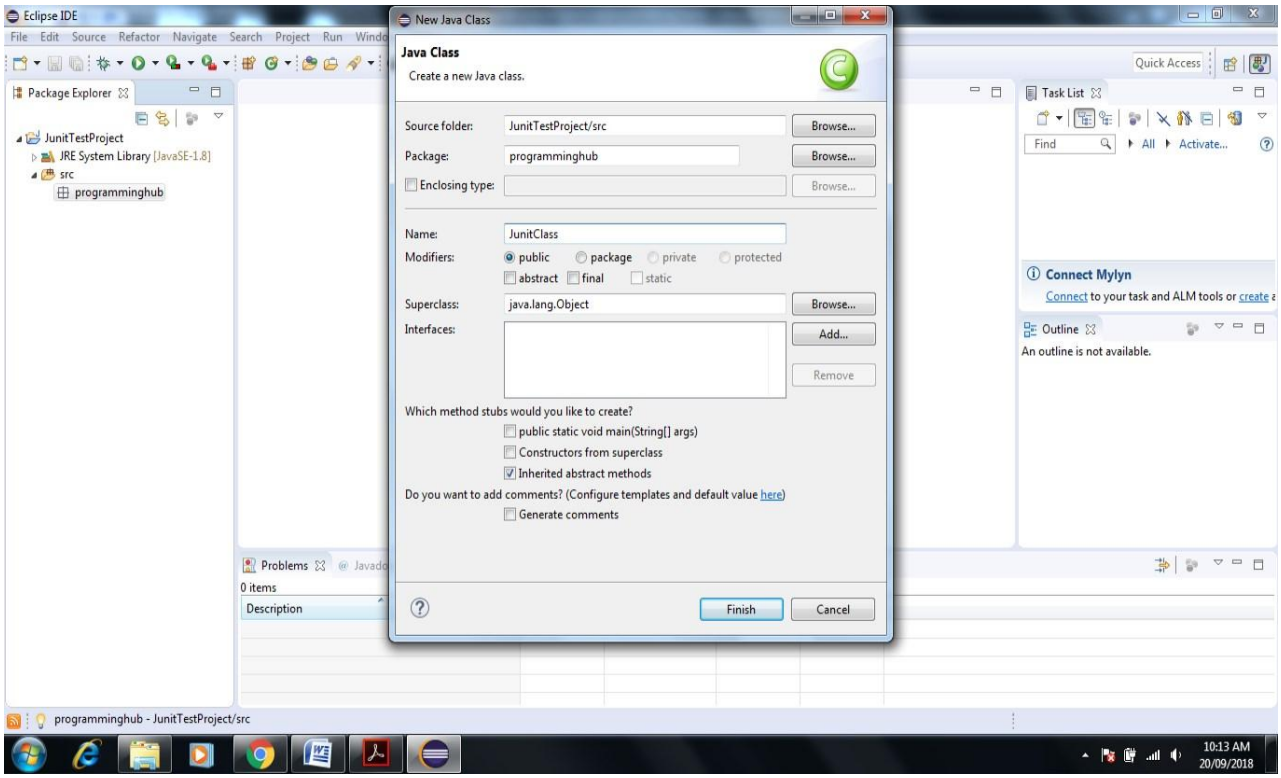
8. Name package as programming hub-> Click on Finish



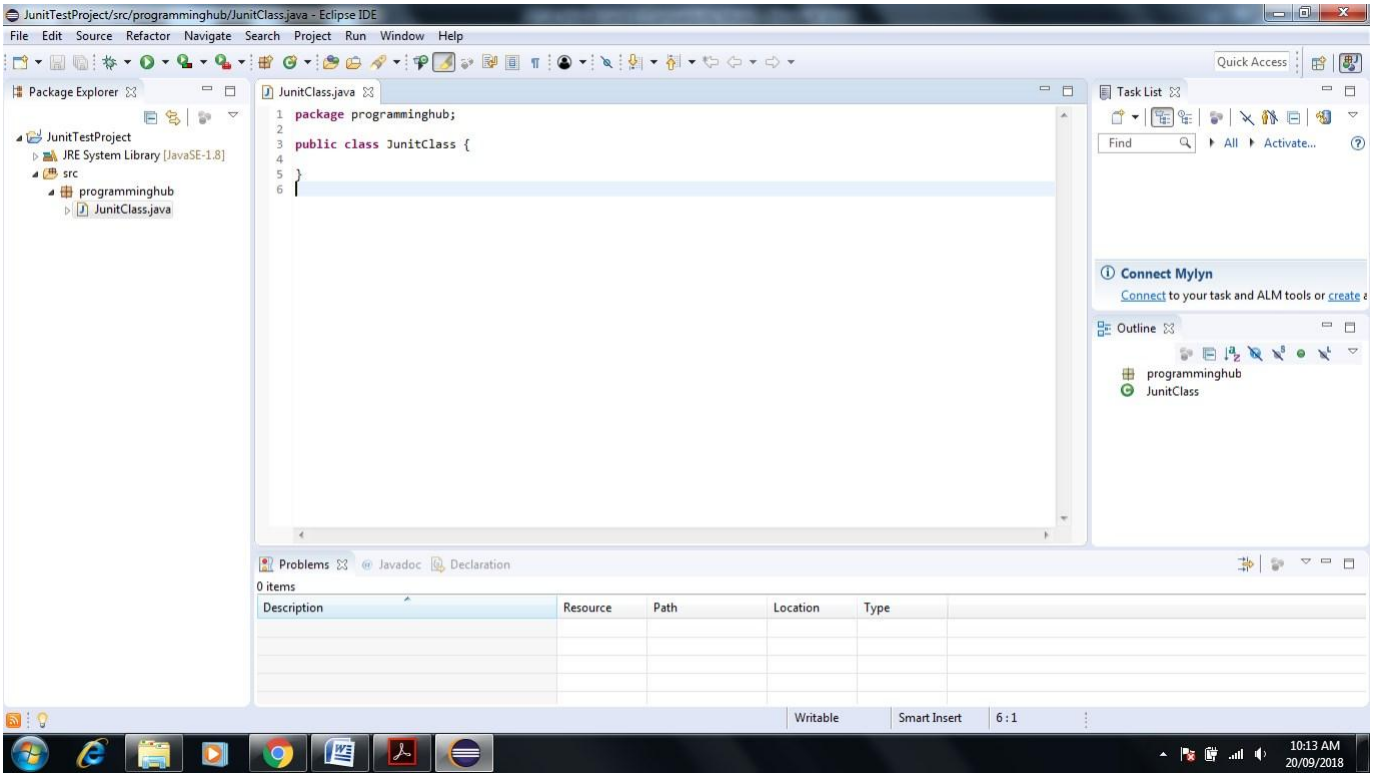
9. See the Programming hub package see in project Explorer Screen of Eclipse



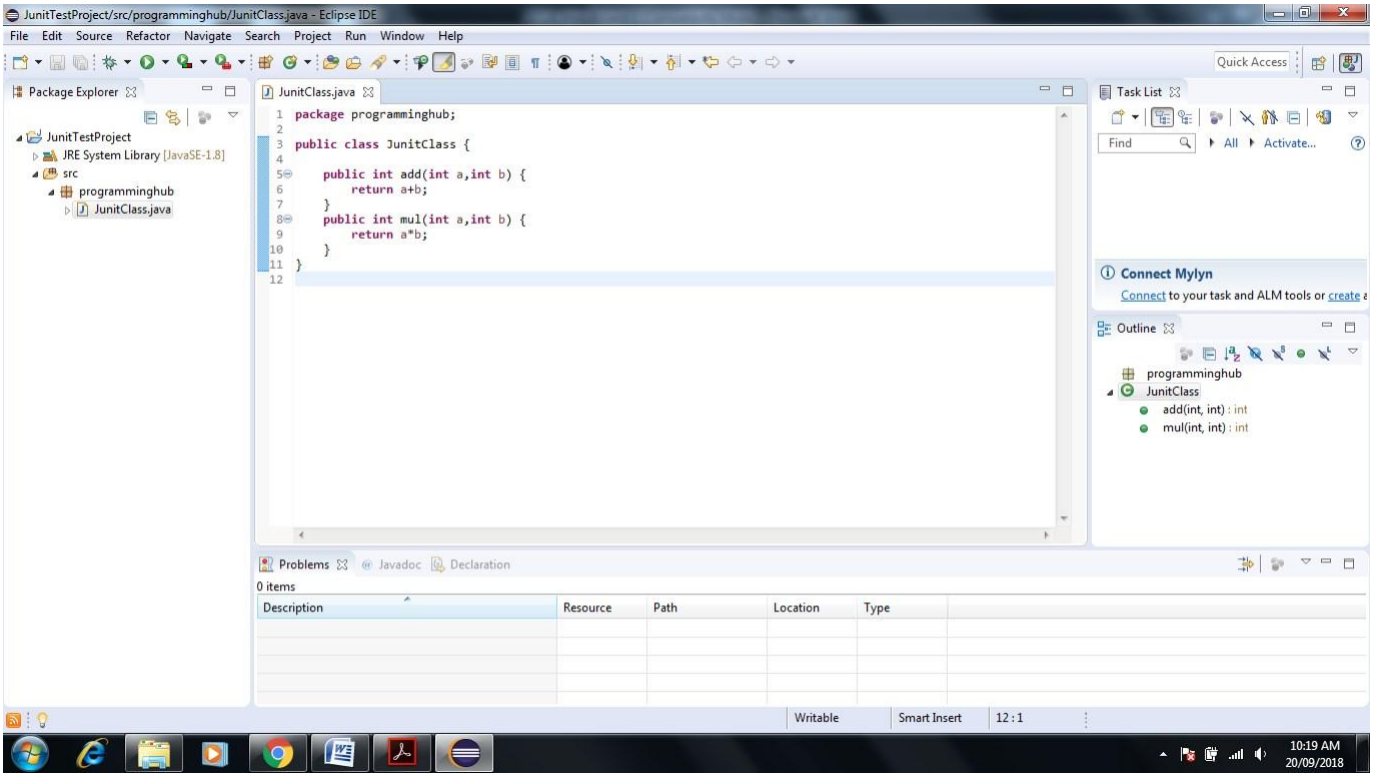
10. Right Click on Programminghub Package->New->Class give the name JunitClass->Click Finish.



11. Next screen will appear

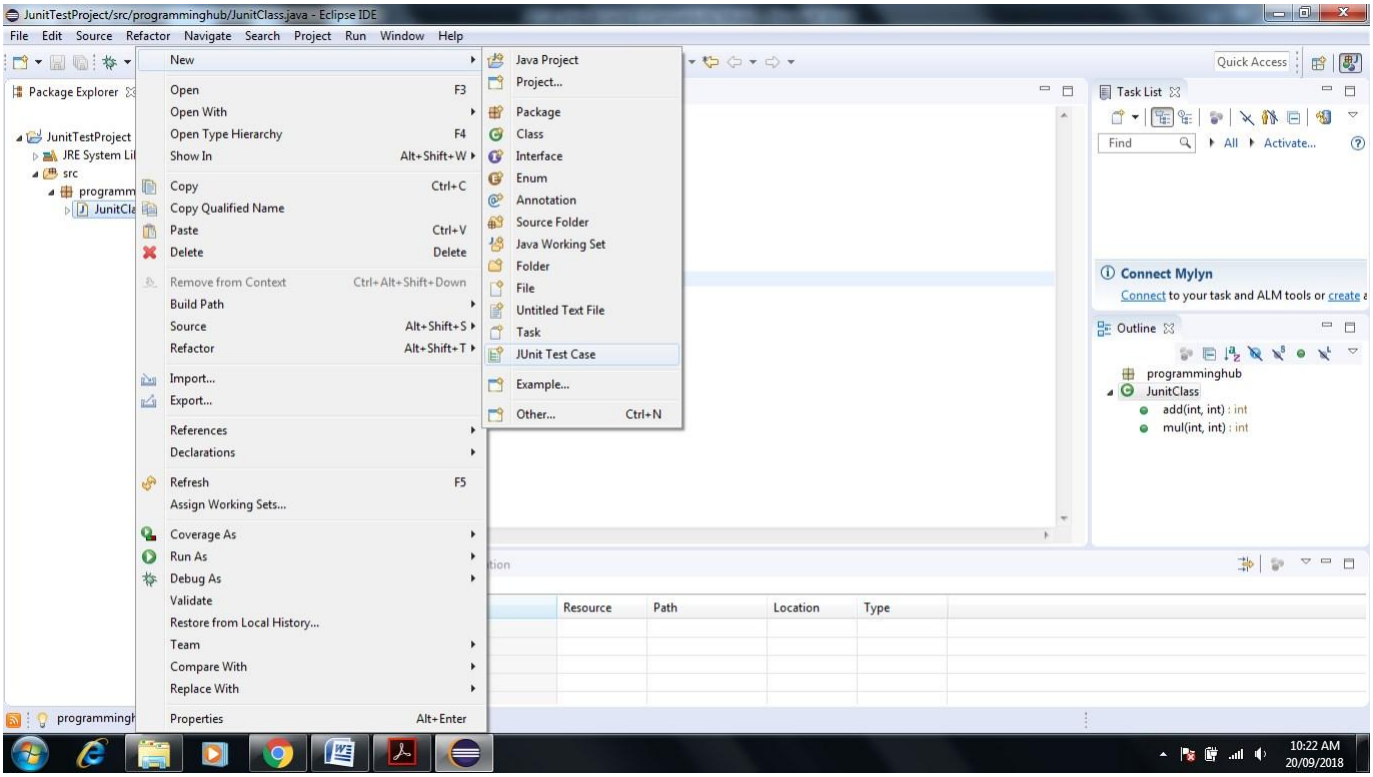


12 Write a small program with only two functions Add and Multiplication

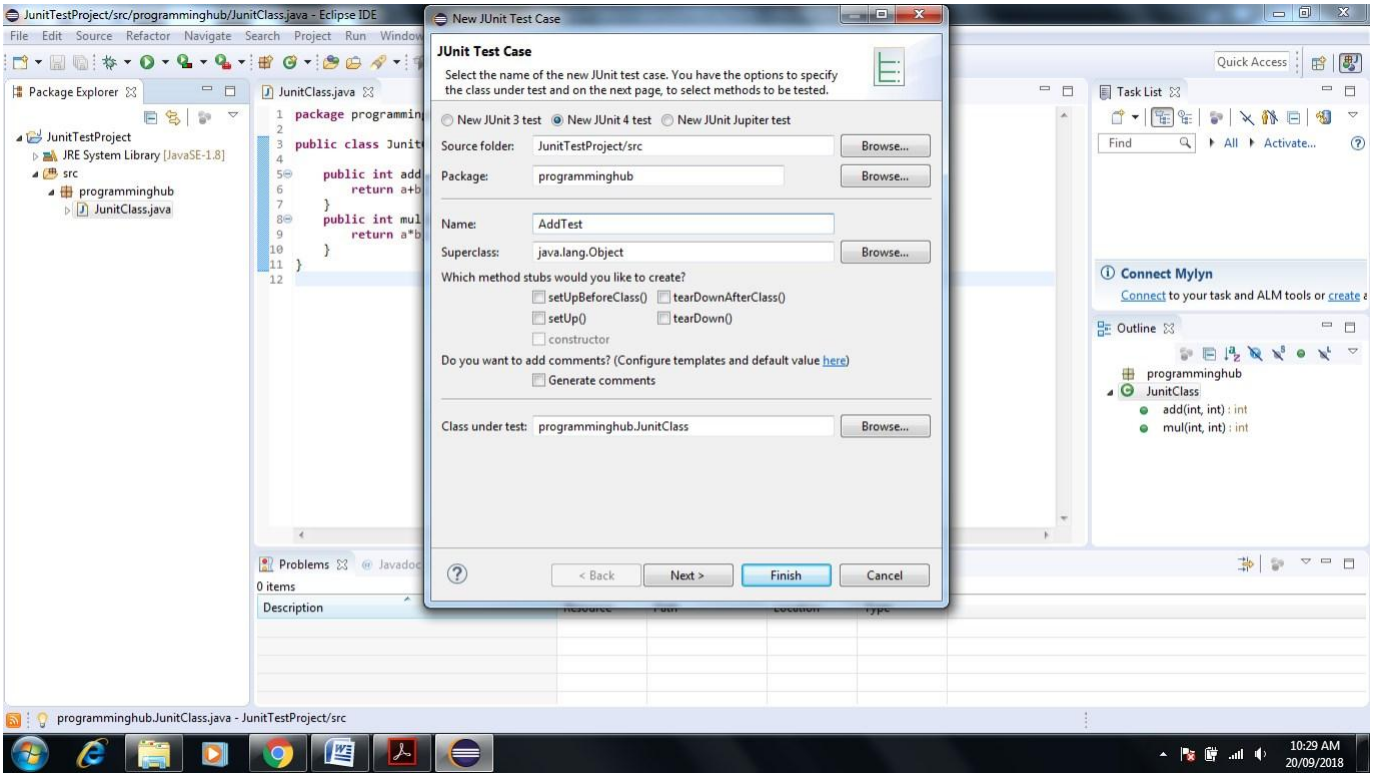


13. Write Test Cases for Java Program

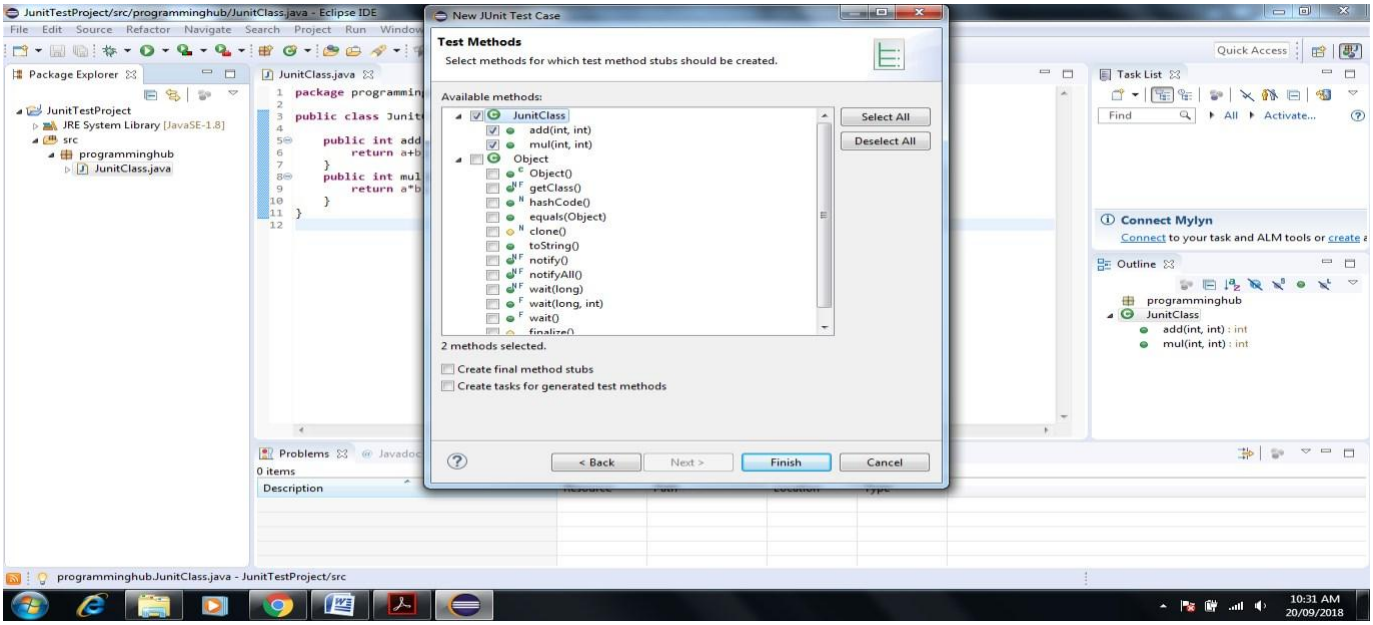
Right click on Junitclass-> New-> Click on Junit Test Cases



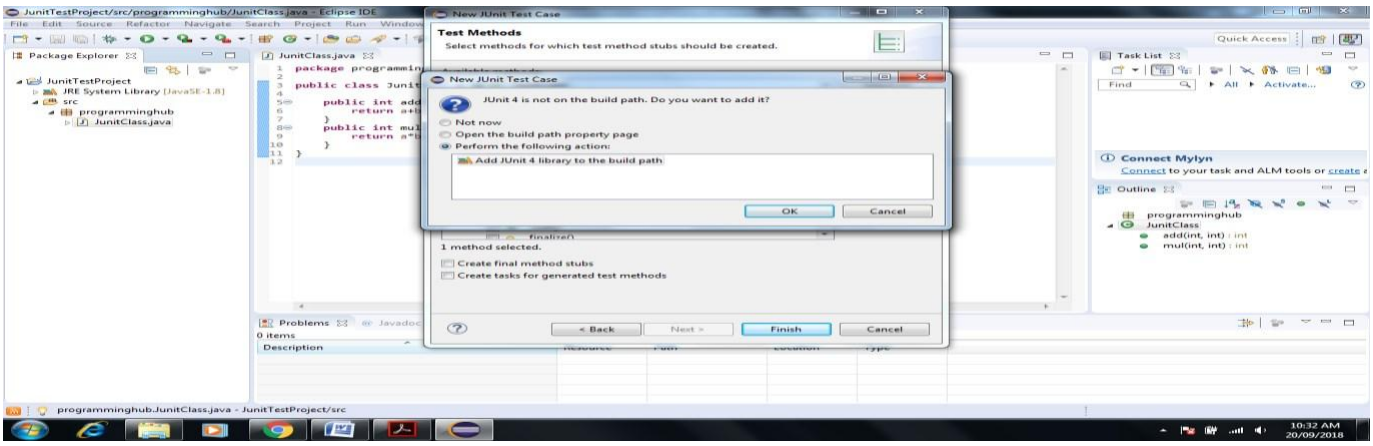
14. Name test suite as AddTest and choose New Junit4 test



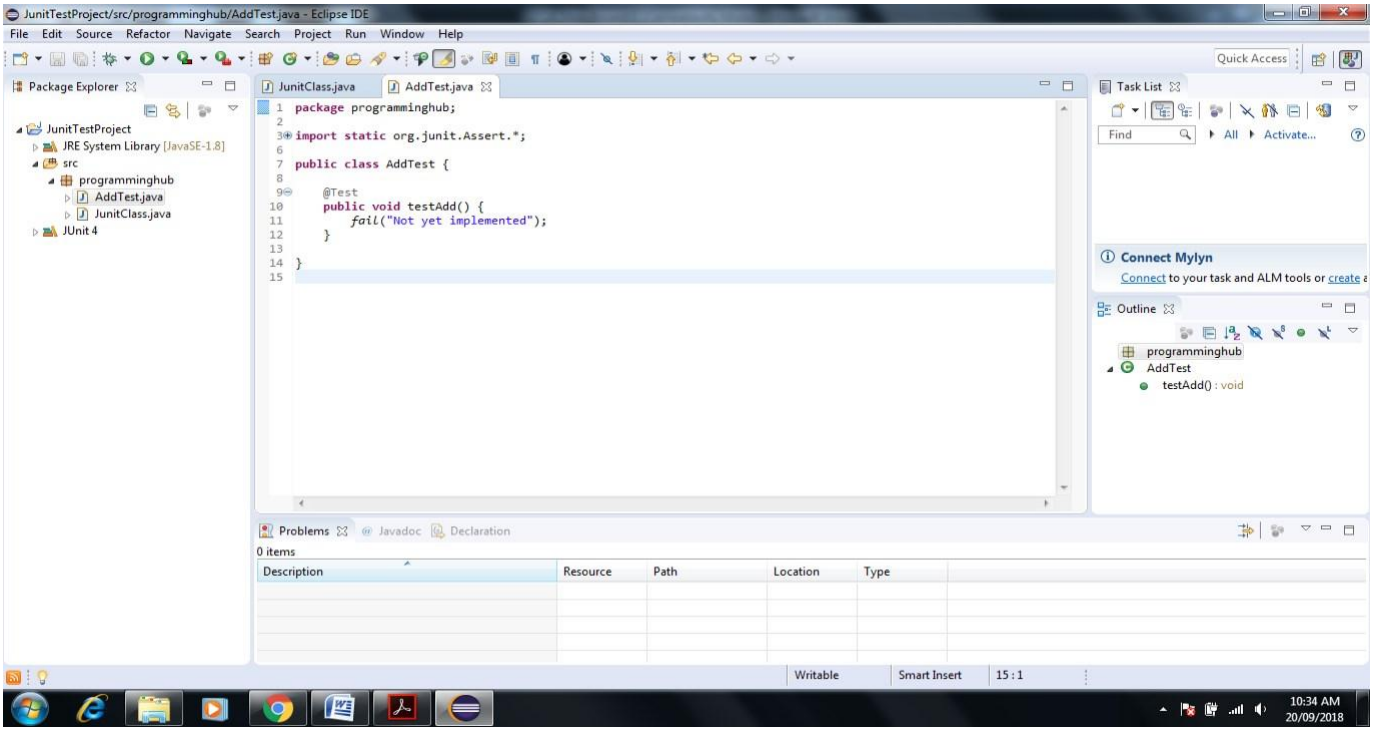
15. Click on add Checkbox



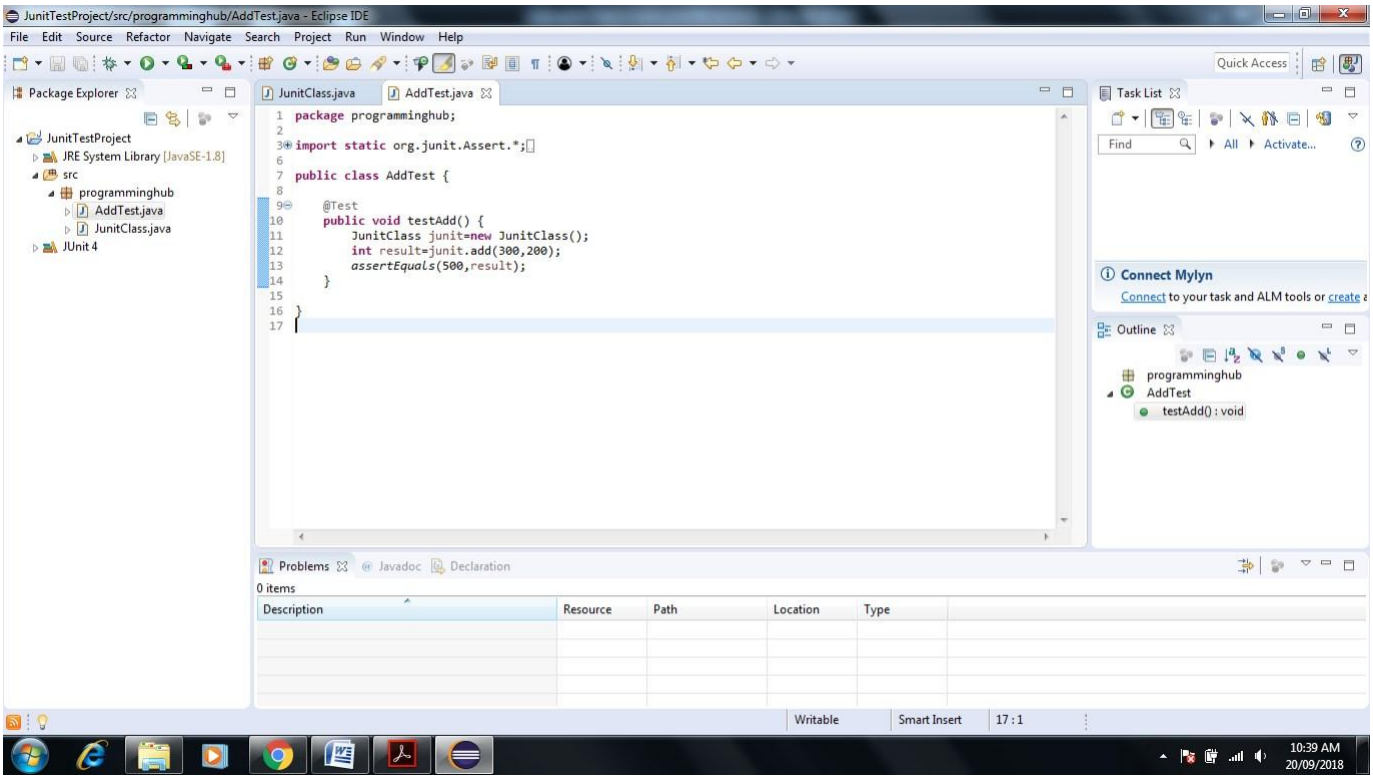
16. Click on Next-> Ok



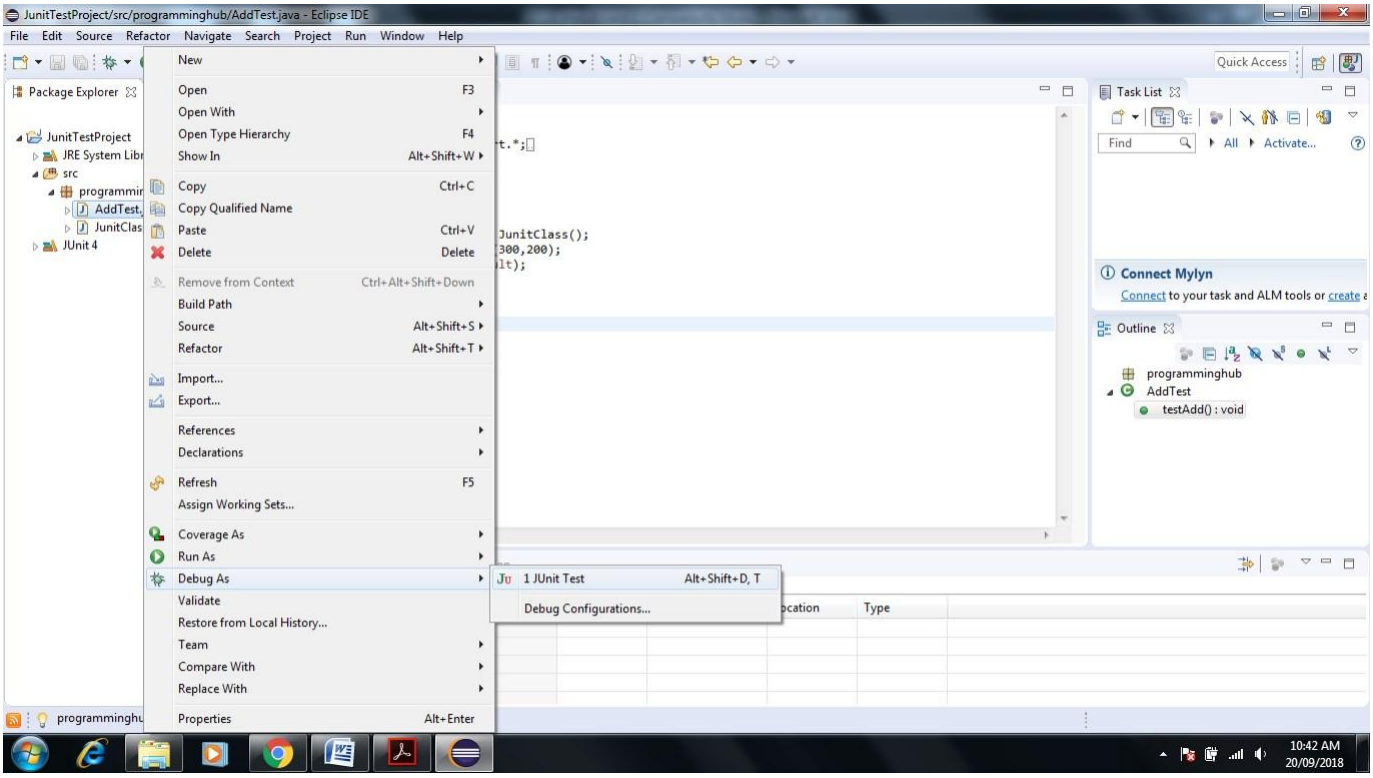
17. Next screen will appear



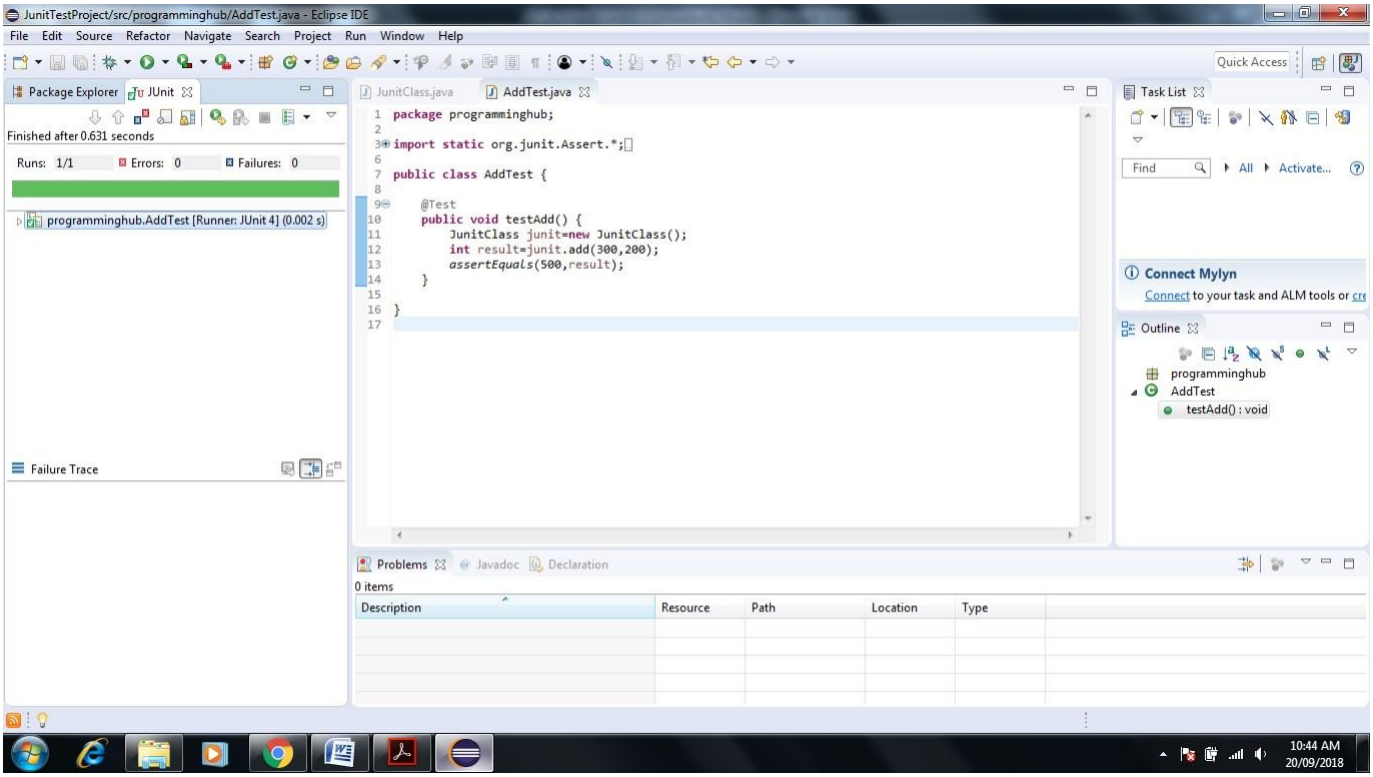
18. Write a code for Test case addition of two number inside AddTest



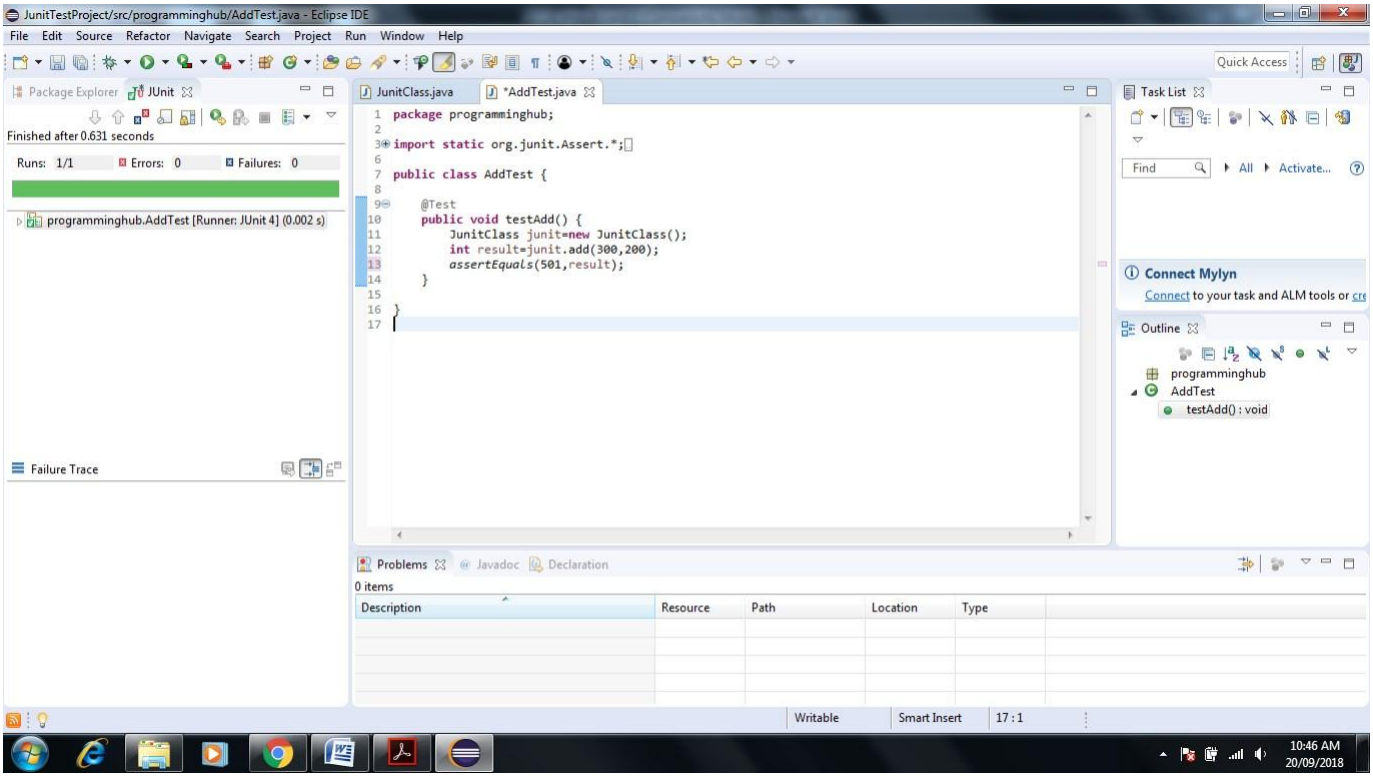
19. Let us run AddTest test case. Right click AddTest-> Debug As->JUnit Test



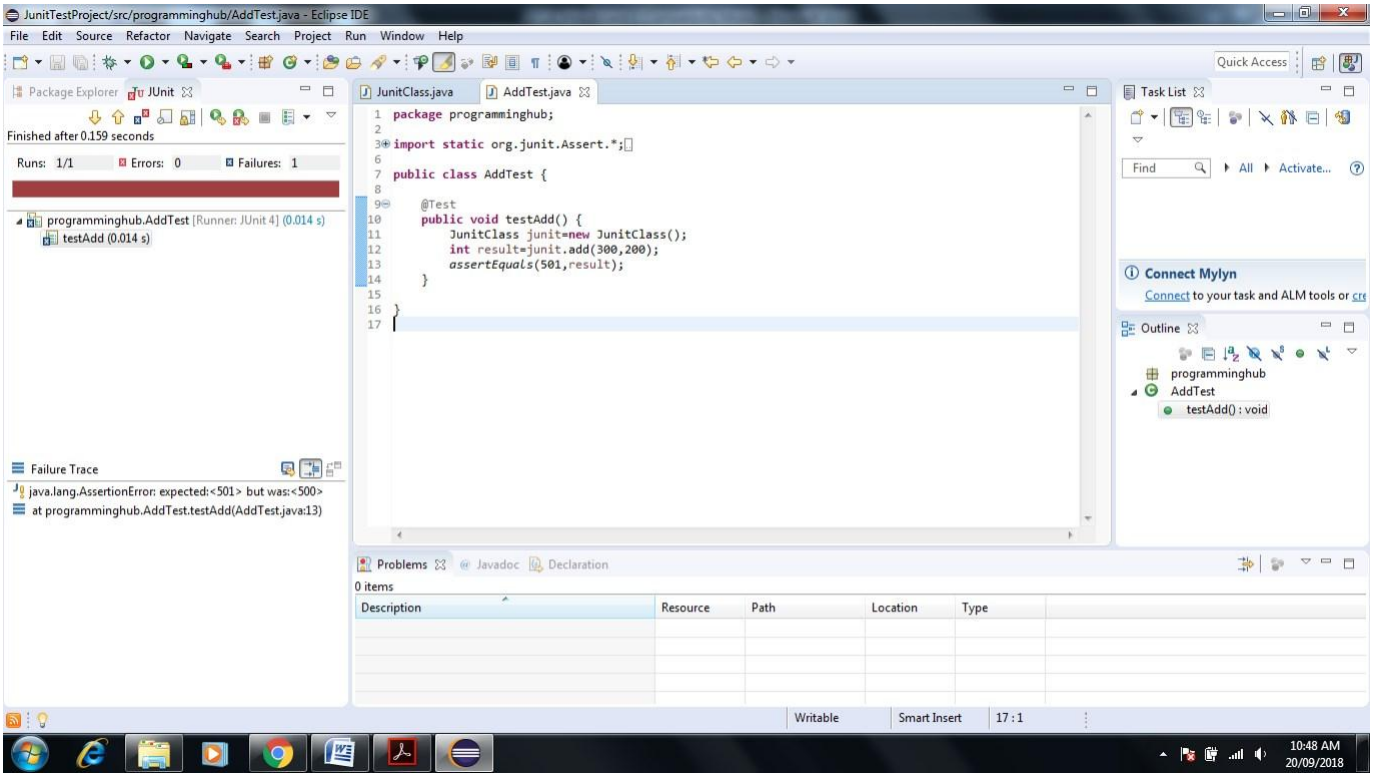
20. Result of test case is as follows. It shows 0 error and 0 failure and green color test bar which means that test case has run successfully(Green Color Bar Indicate)



21. Let us purposely give wrong input in assertEquals method or unexpected result here we write 501 instead of 500 indicate wrong addition result

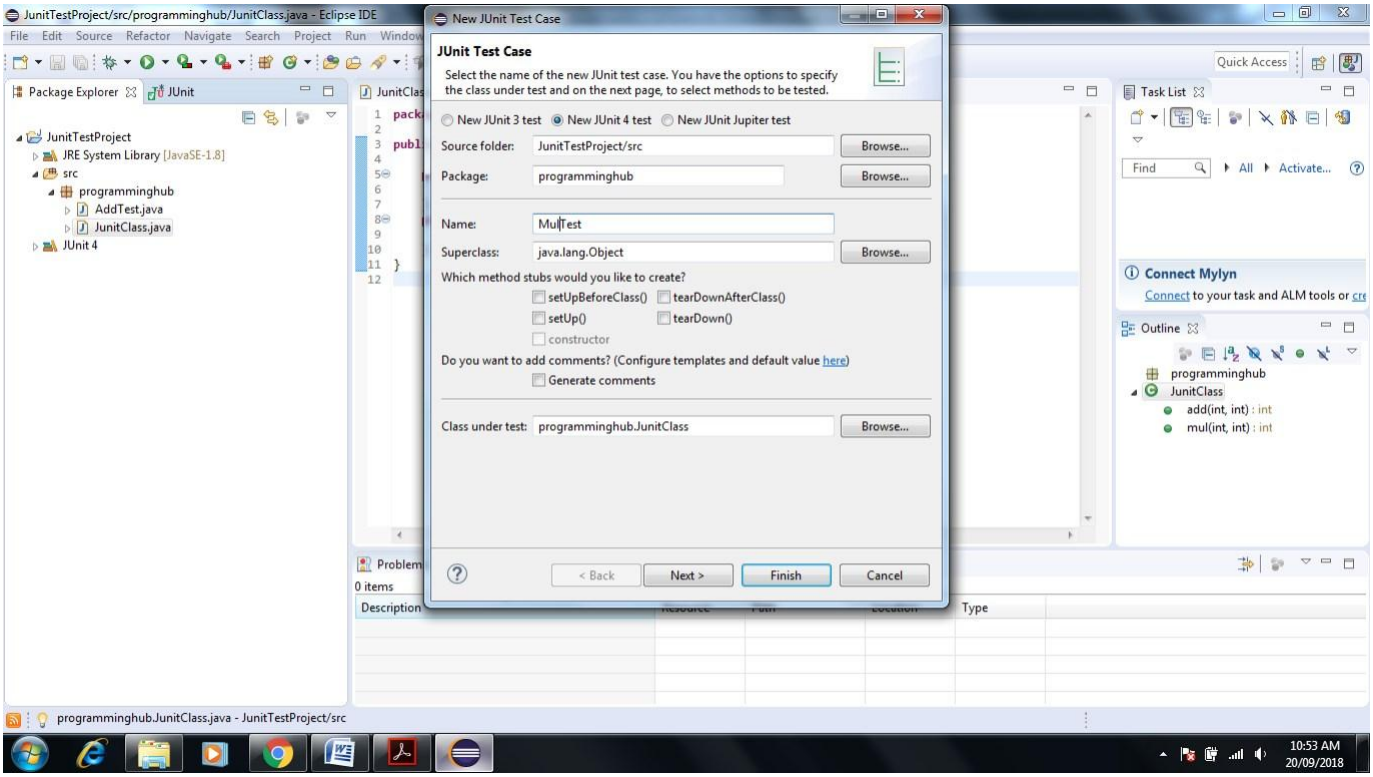


22. Now test case should fail.(Brown Color Bar Indicate) So again run AddTest as follows

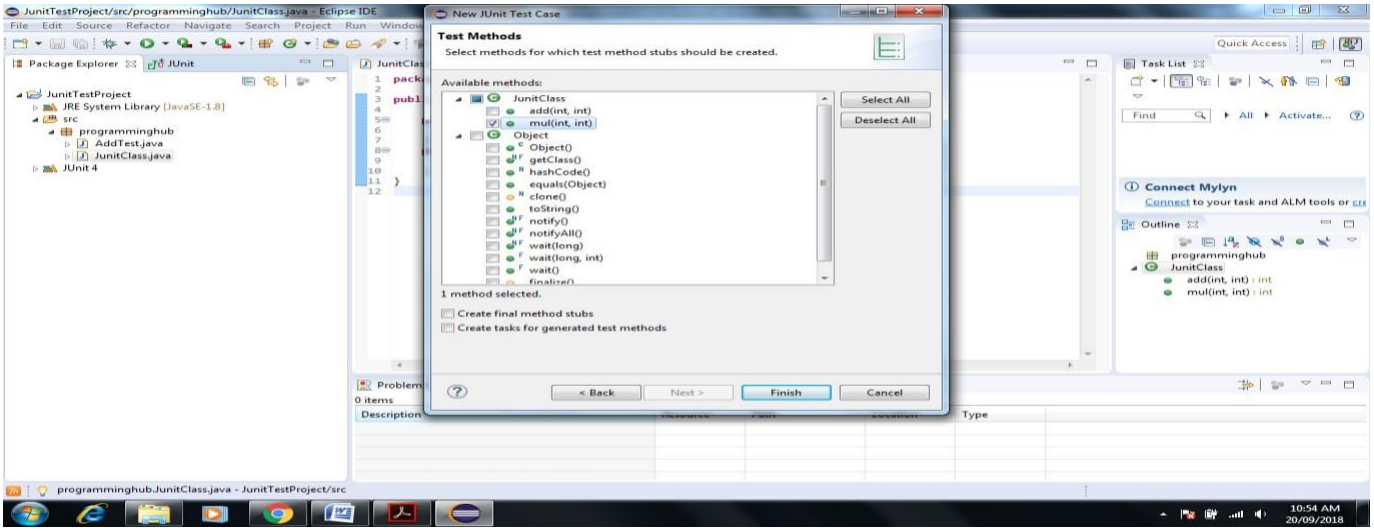


23. Similarly you can Create Test case for Multiplication Function

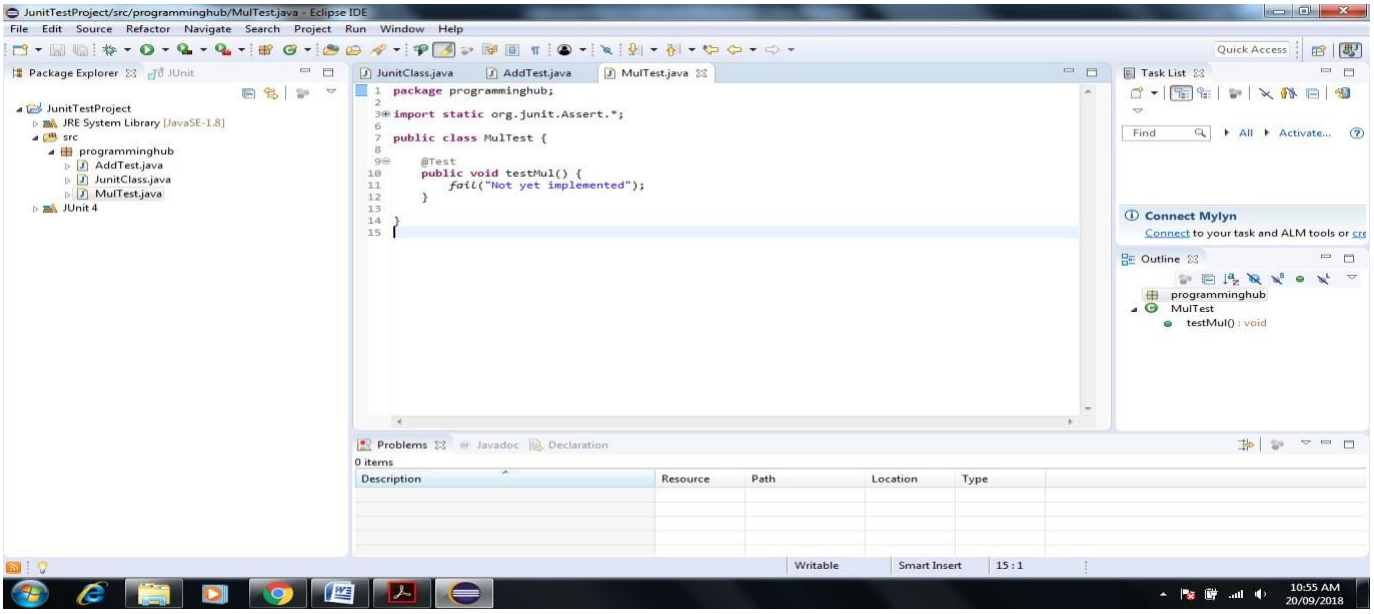
Click on Project Explorer Screen-> Right Click on JunitClass->New->JUnit Test Case-> Give name MulTest.



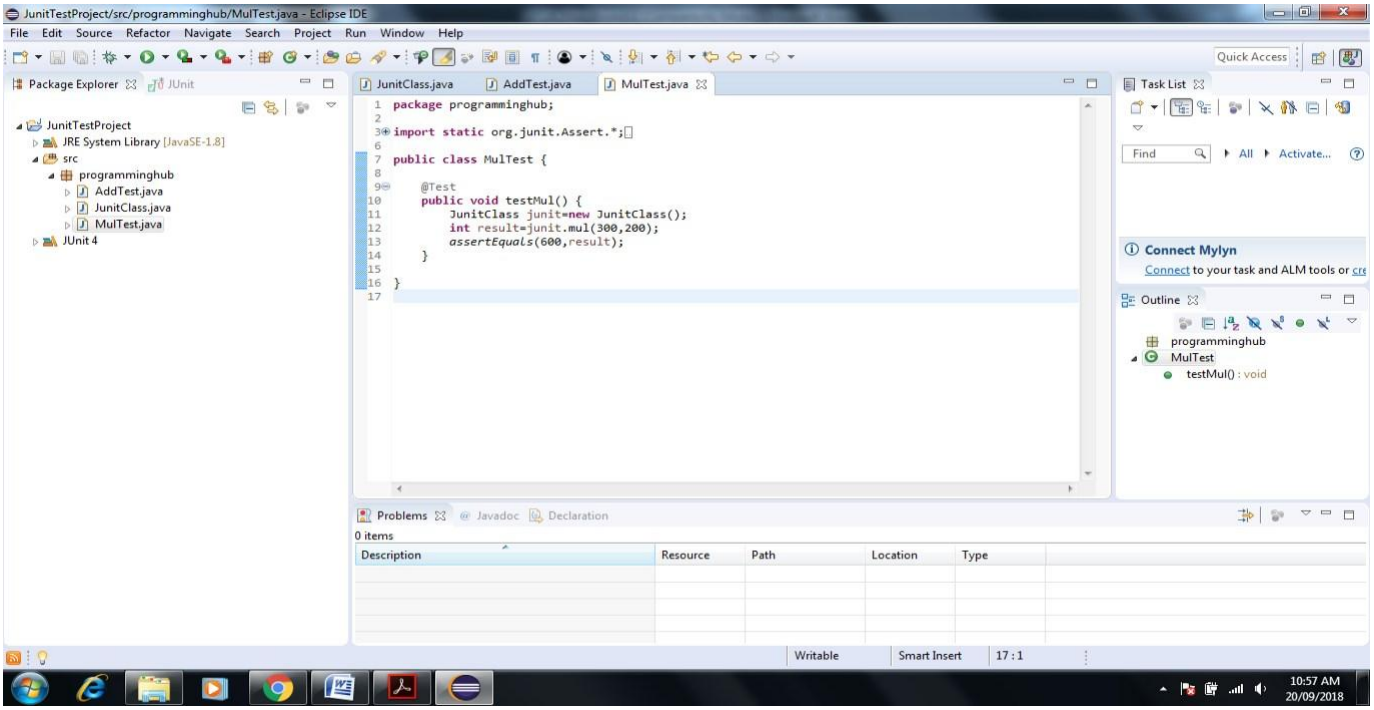
24. Click on Next ->Select Mul Check Box -> Click Finish



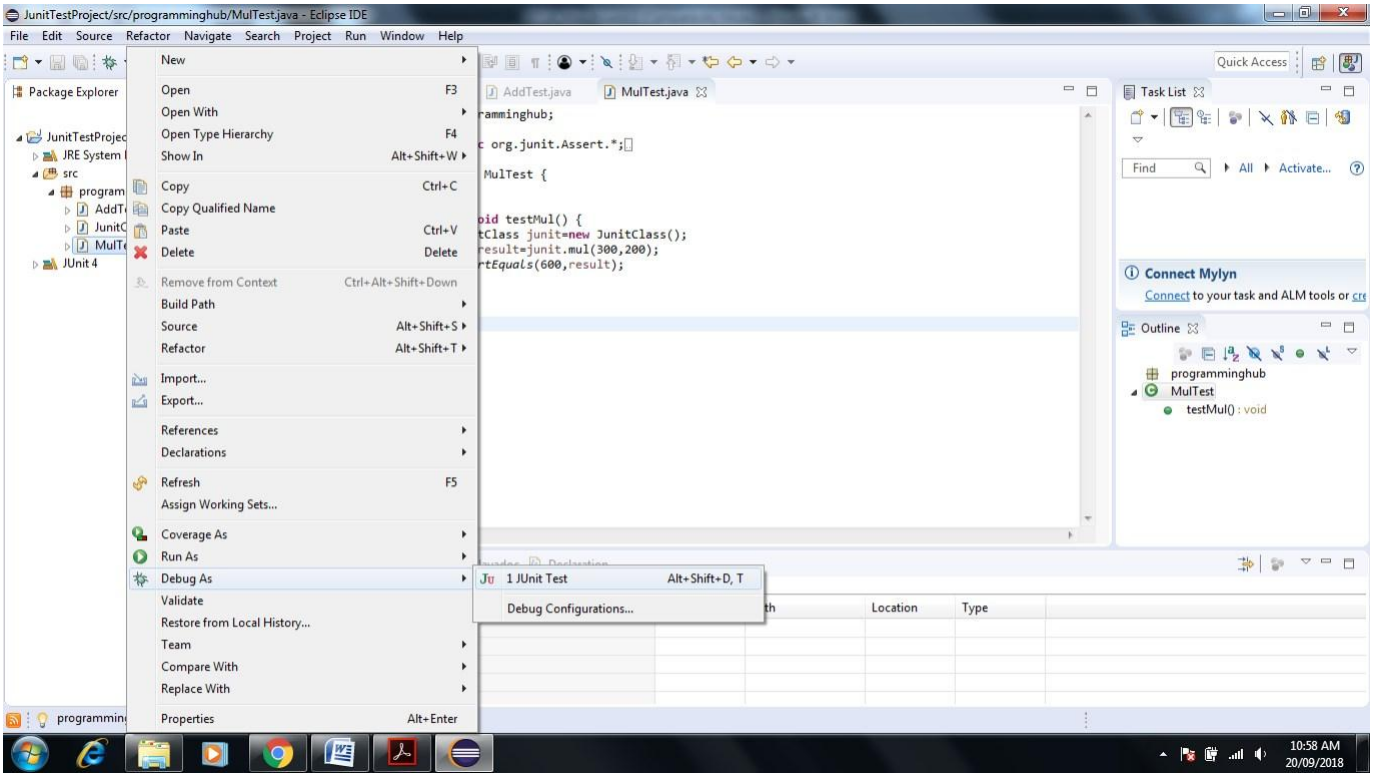
25. Next Screen will appear



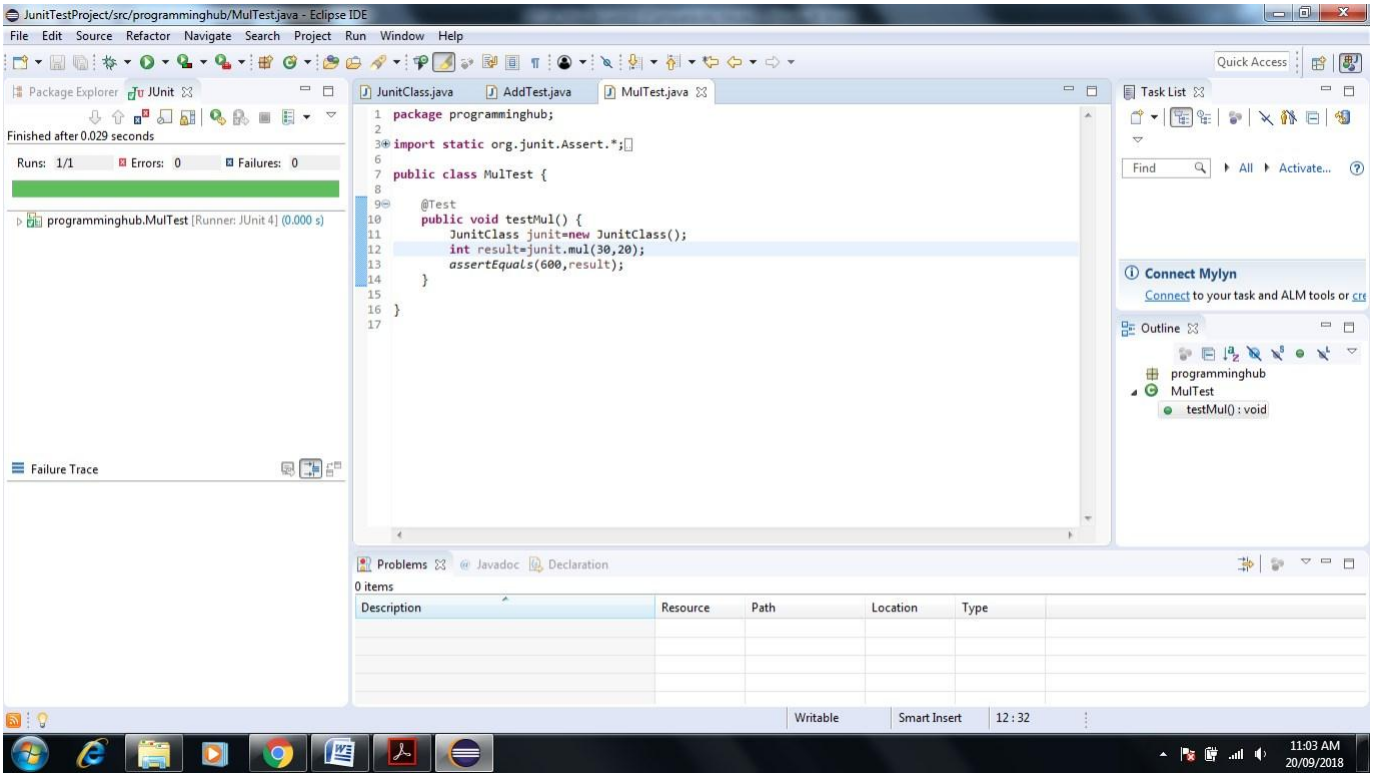
26. Write a Test Case Code inside MulTest method



27. Right Click on MulTest->Debug->JUnit Test

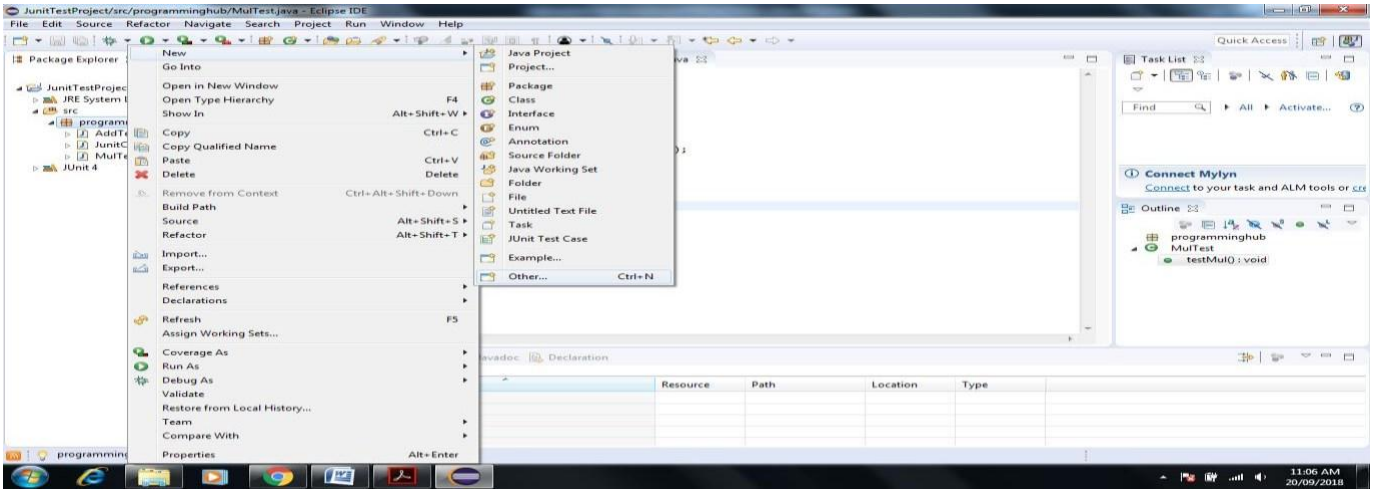


28. Execute Test

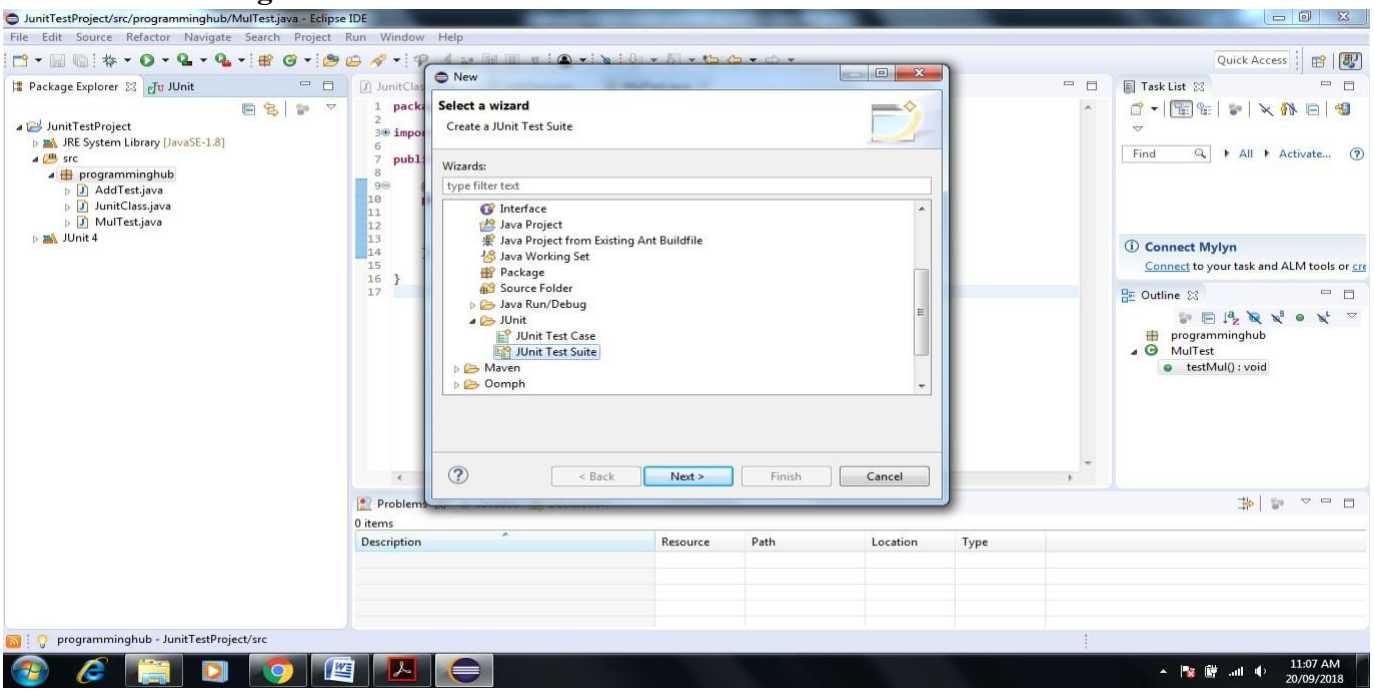


Test Suite – it is used to test multiple test cases at one time.

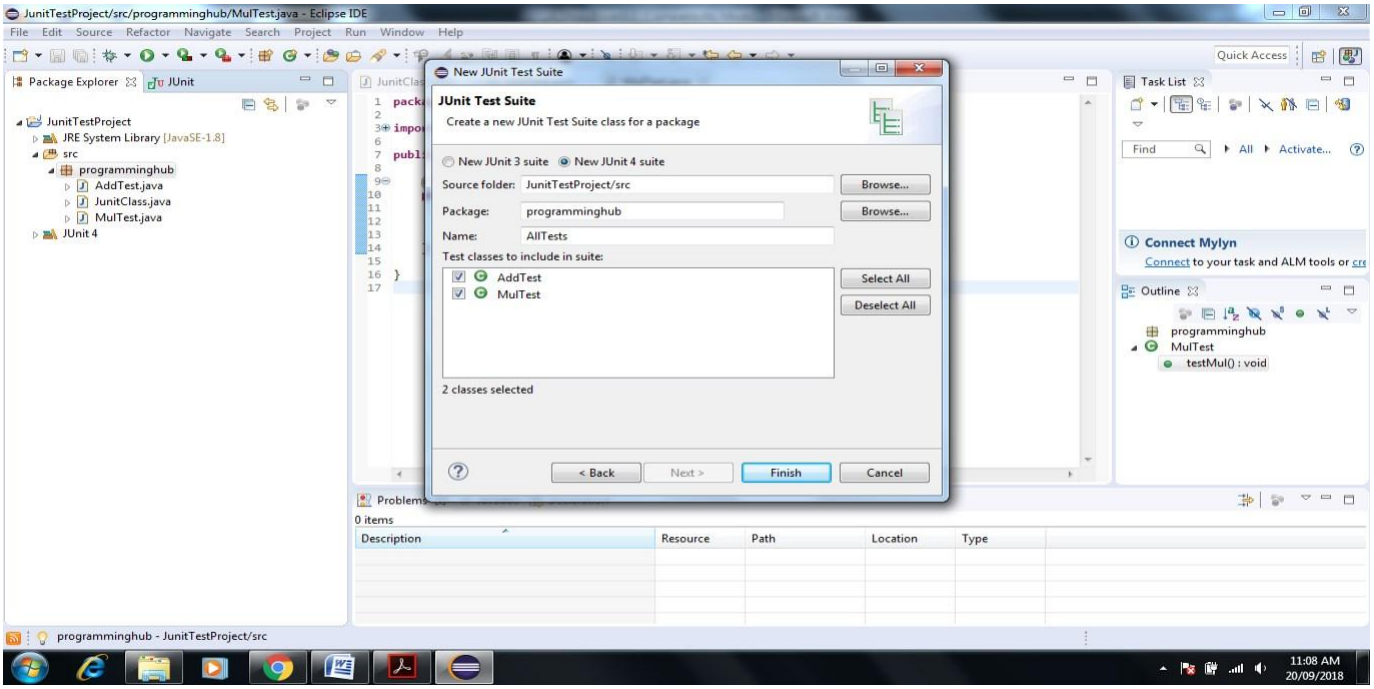
29. Now let us create Test Suite both add and mul test cases in one time



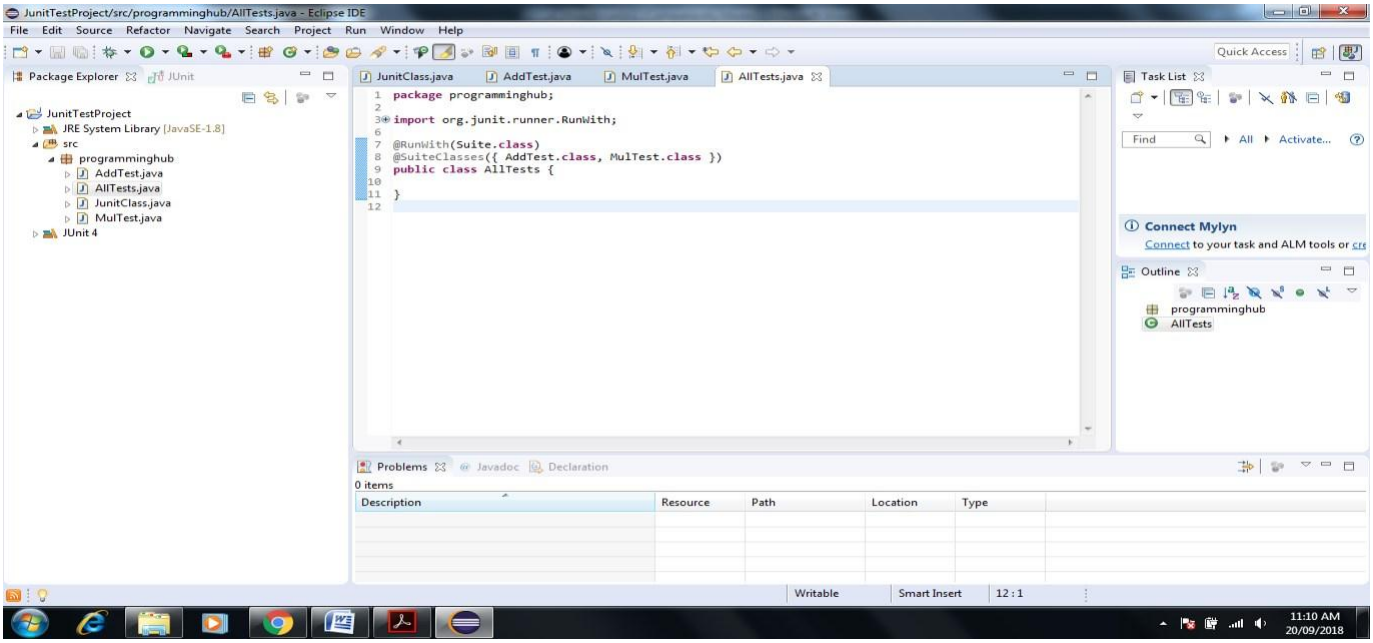
30. Click on Package name->New->Other->JUnit->JUnit Test Suite->Next



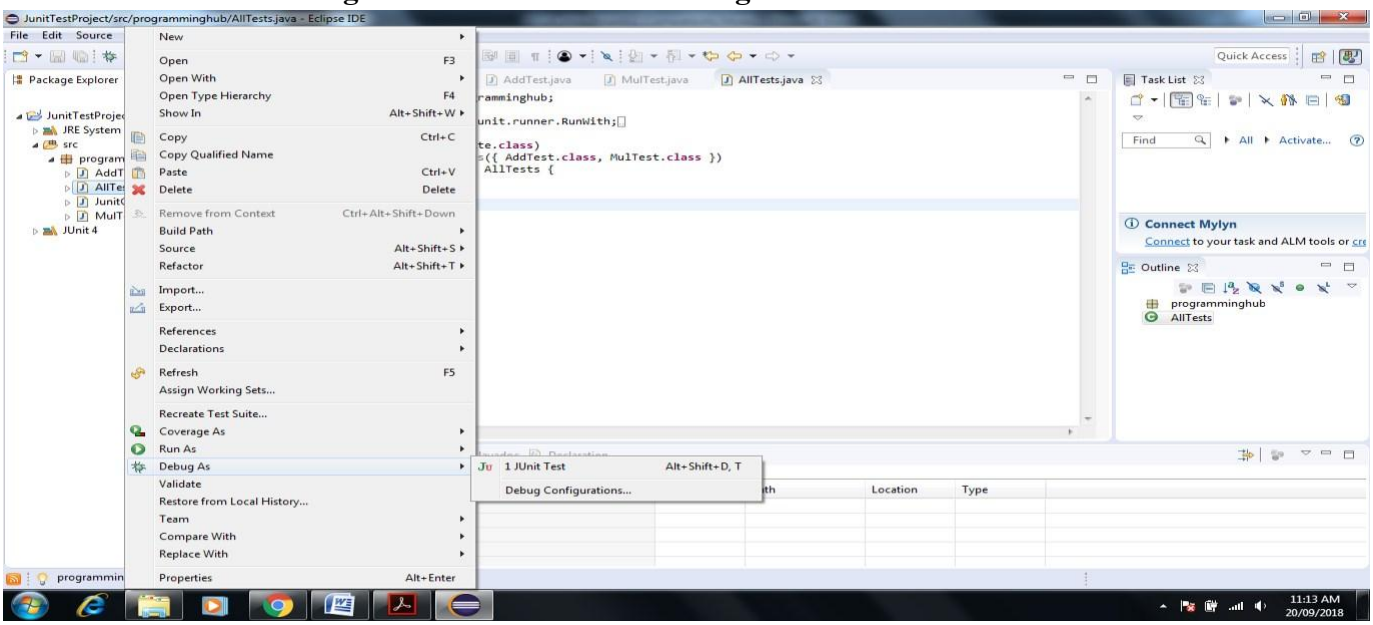
31. Click on Finsh



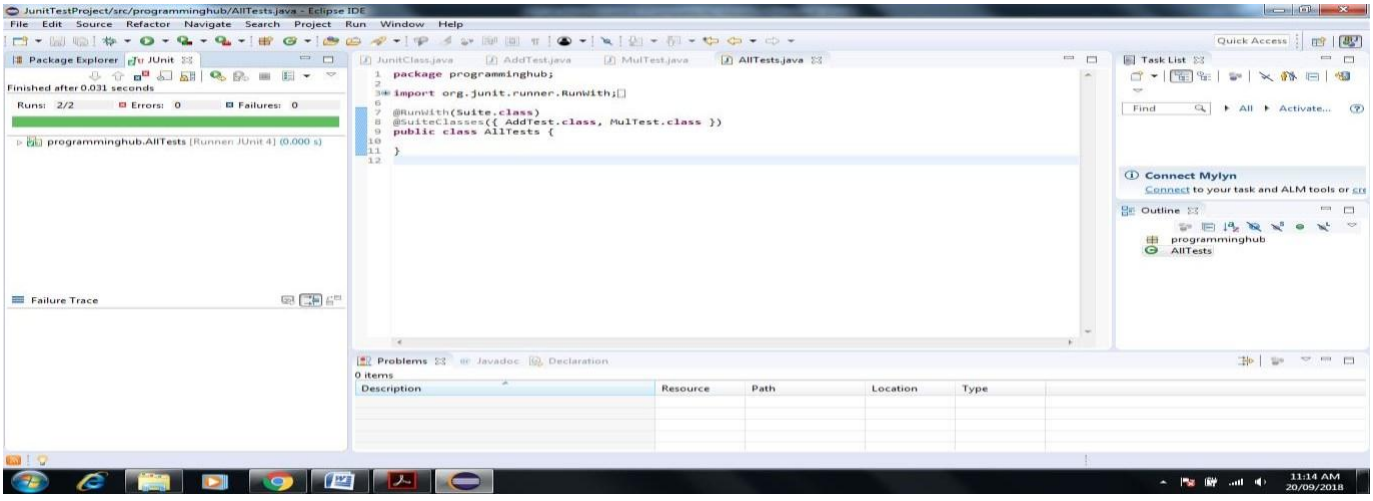
32. Next Screen Appear that automatically create Test Suite for Add and Mul



33. Execute Test Suite Right Click on All Test ->Debug->JUnit Test



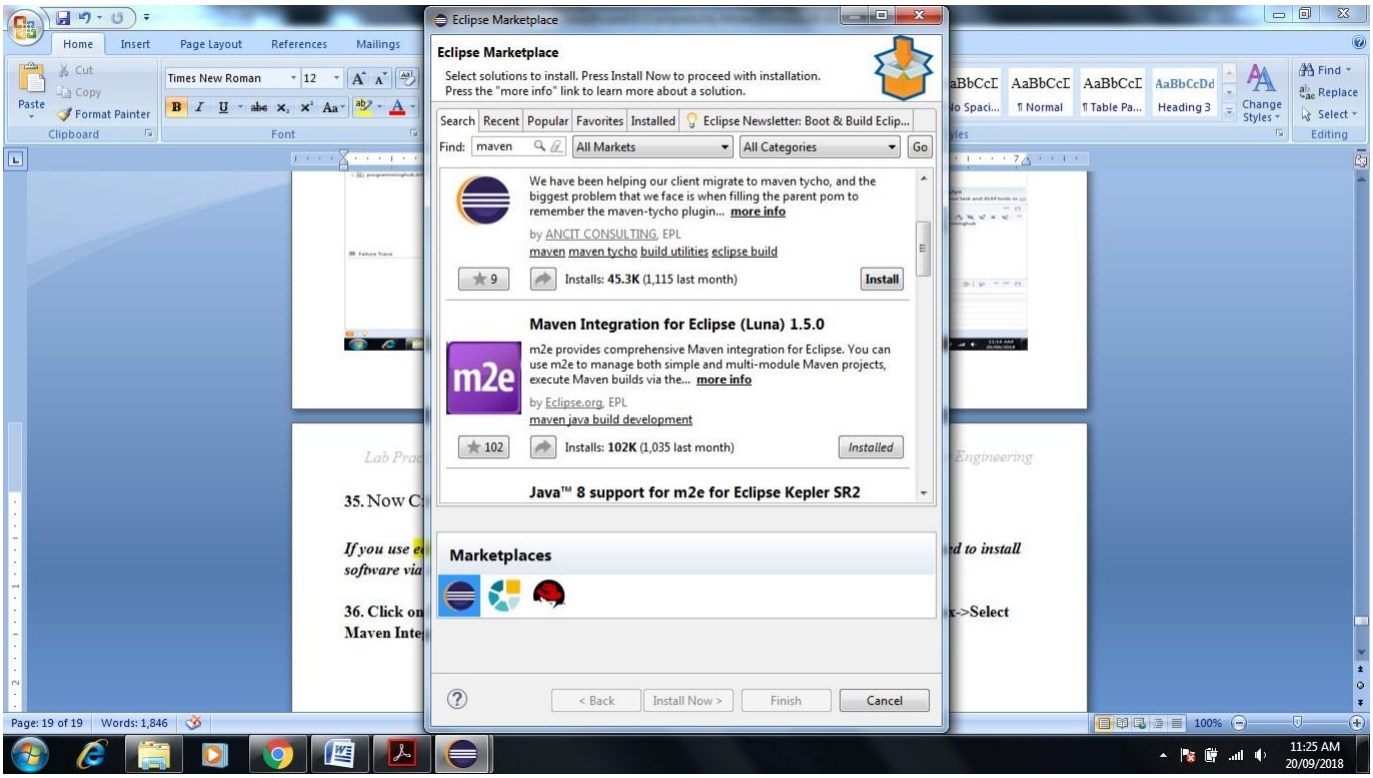
34. Test Suite Executed successfully Test suite fails even if a single test case among all fails.



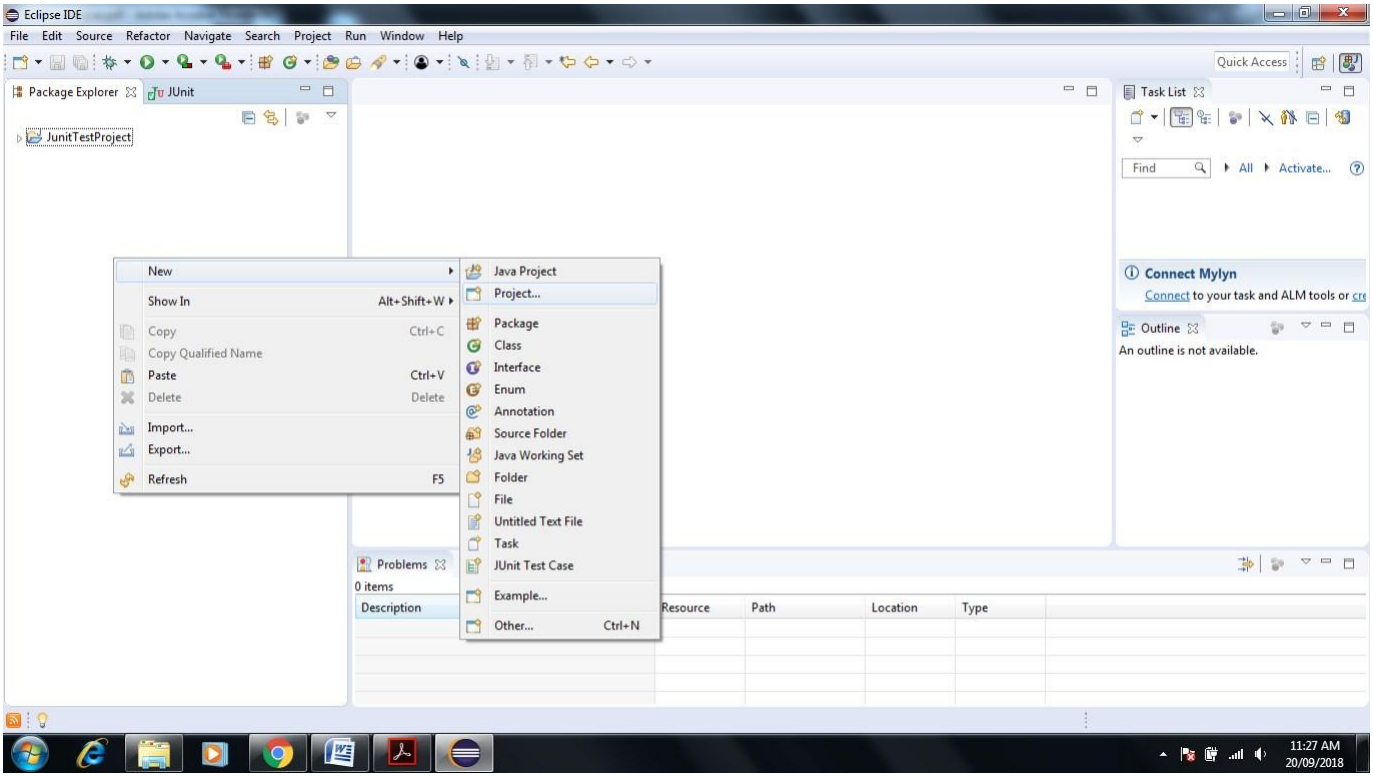
35. Now Create **Test Report** Using **Apache Maven**

If you use eclipse-java-photon-R-win32 Version it include Maven in built installed so no need to install software via Eclipse help Install Software Option

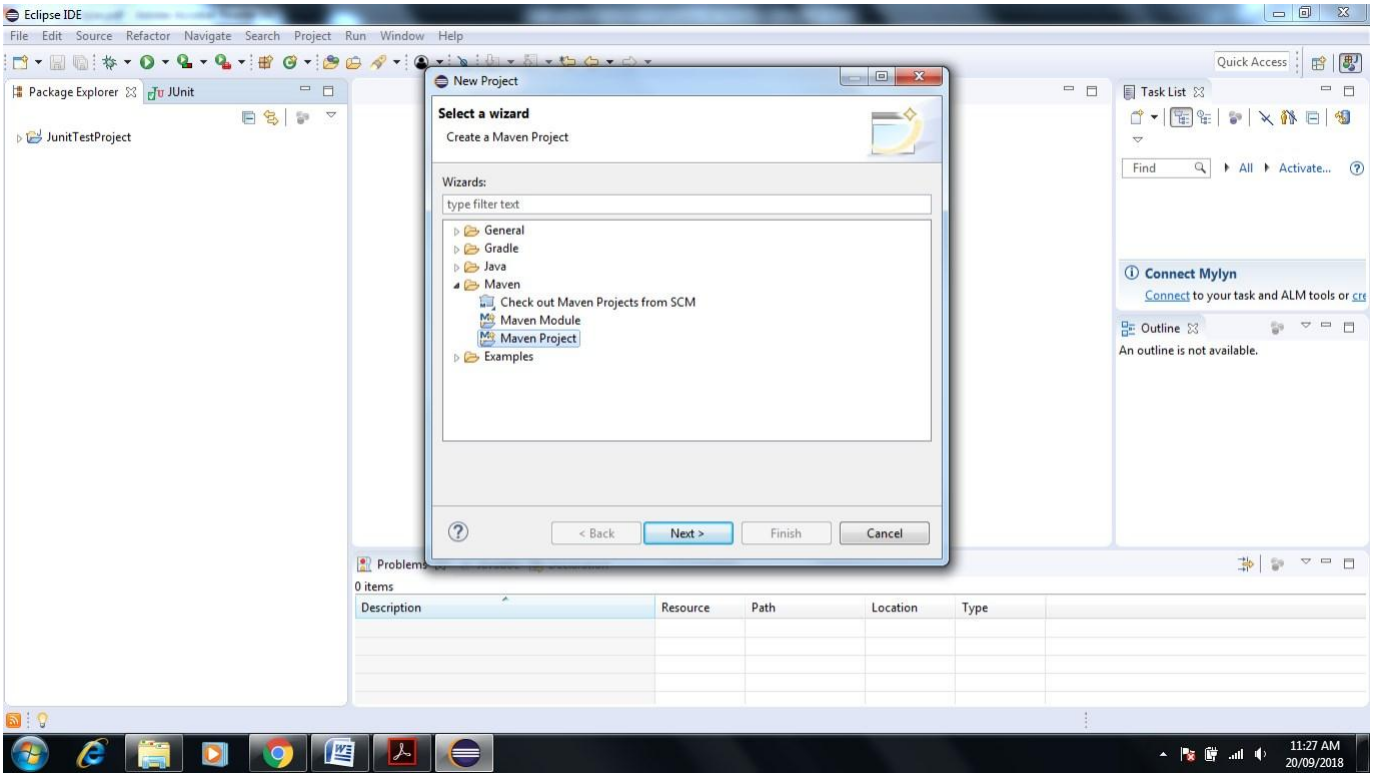
36. Click on Help in Eclipse->Eclipse Marketplace->Enter Maven Keyword in Search box->Select Maven Integration version as per requirement->Click on Install



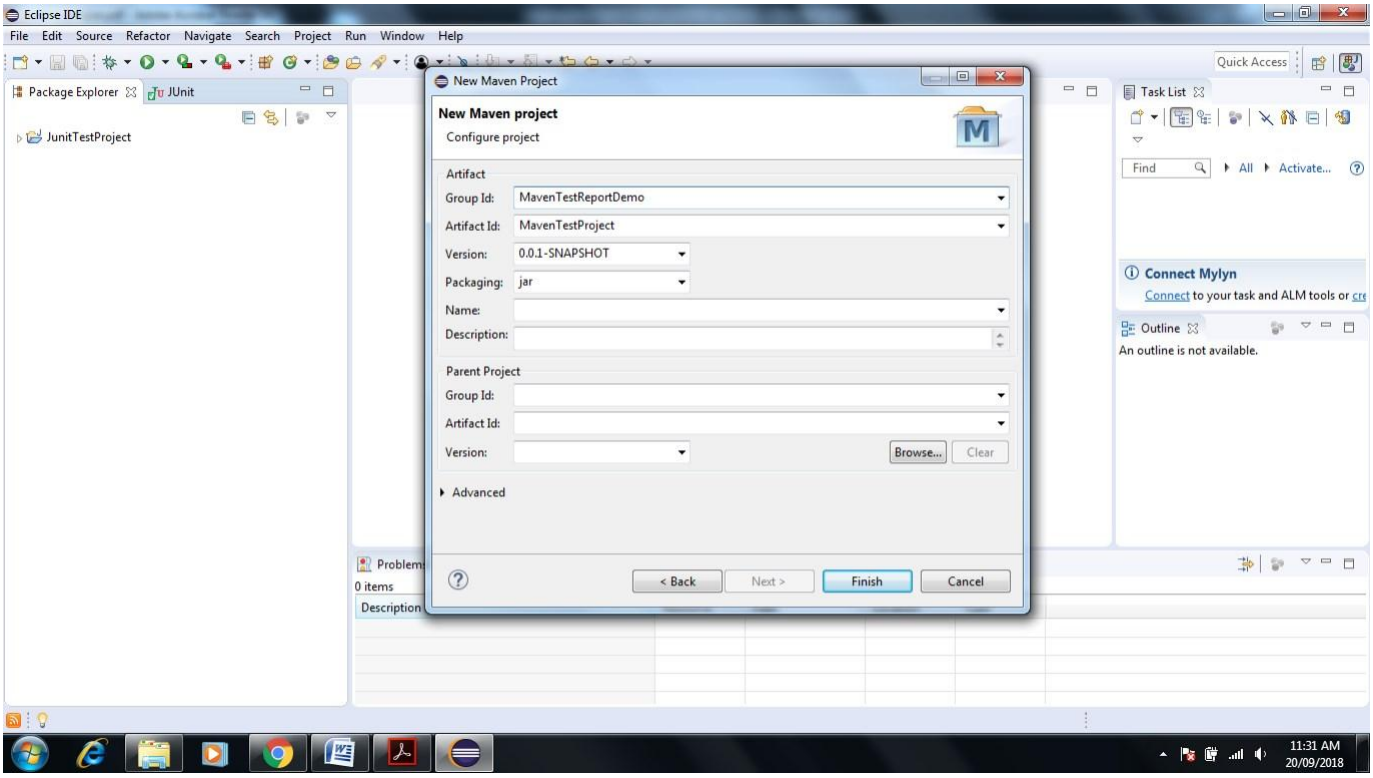
37. Right Click in Project Explorer Window



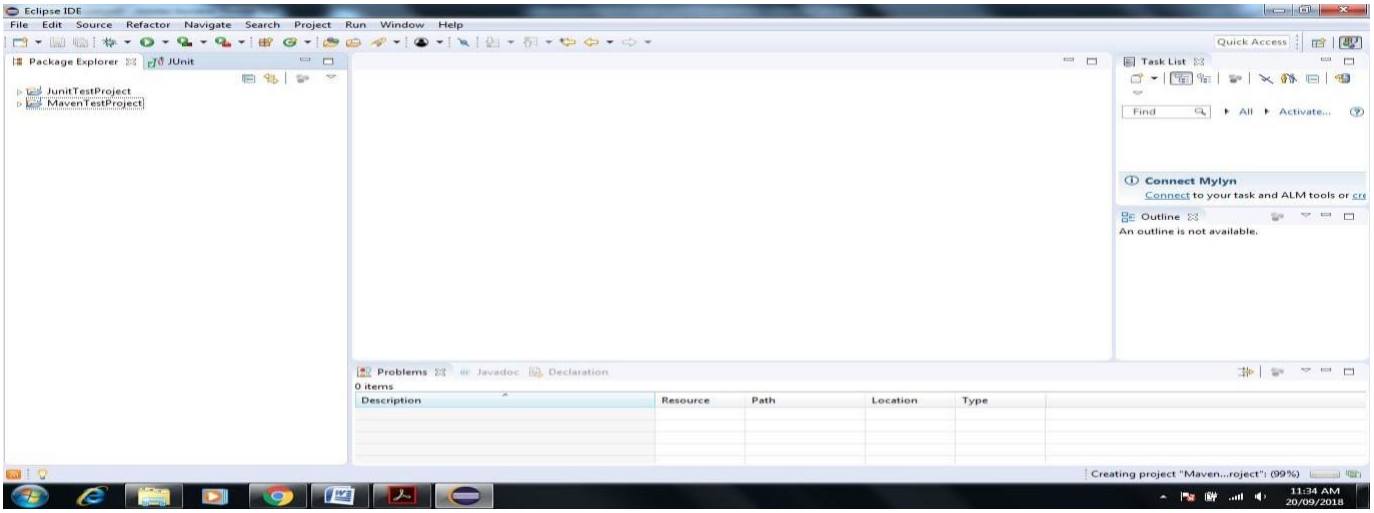
38. Go to Maven Project-> Click Next



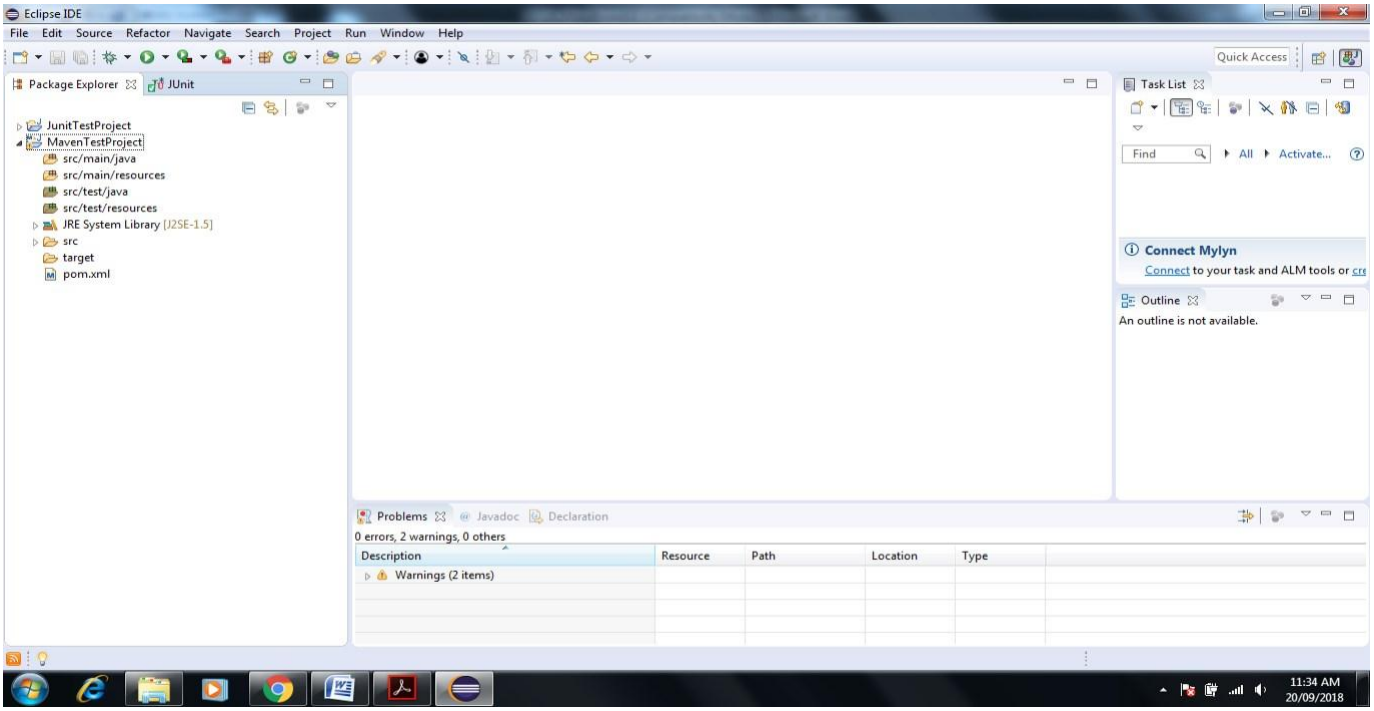
39. Select Check Box Create Simple Project-> Click Next-> Give Group Id and Artifact name



40. Click on Finish-> Next Screen Appear

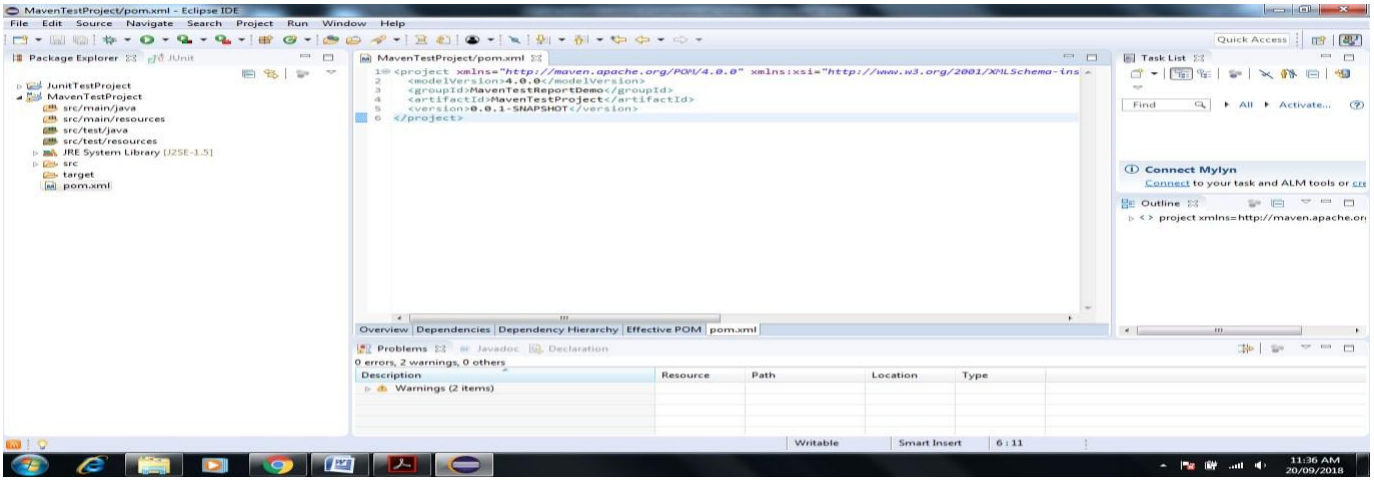


41. MavenTestProject shown Pom.xml file doble click on same

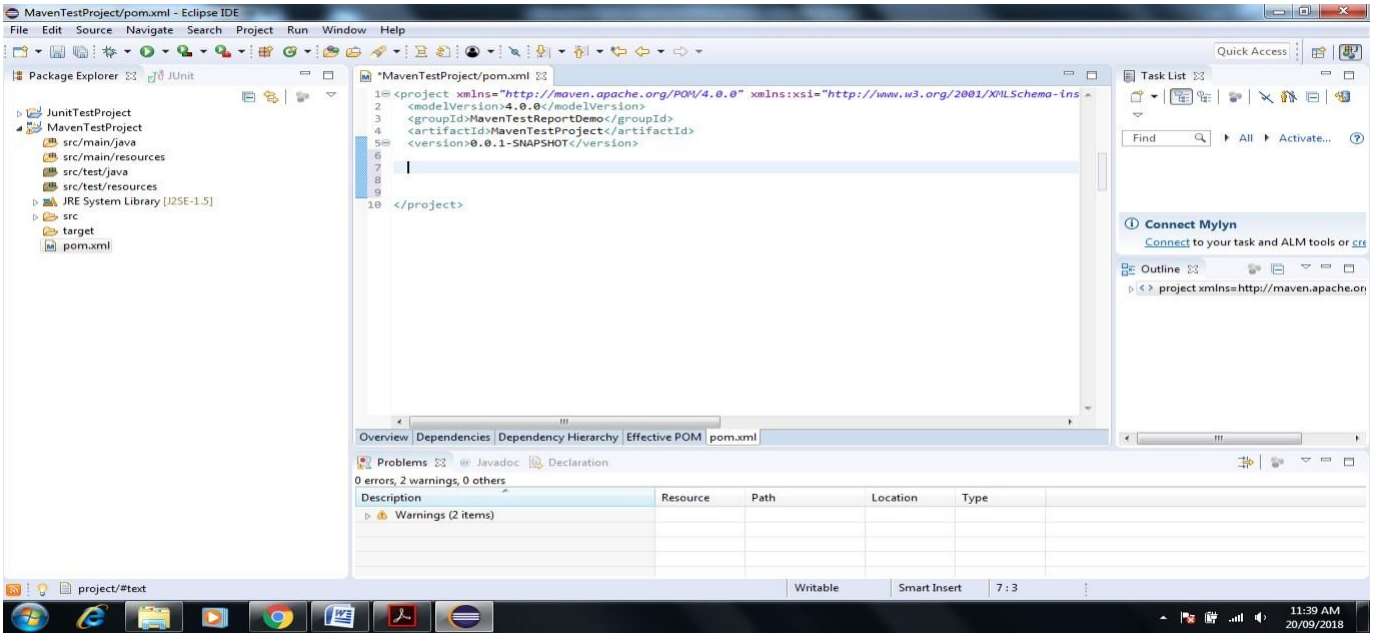


42. it shown some description like

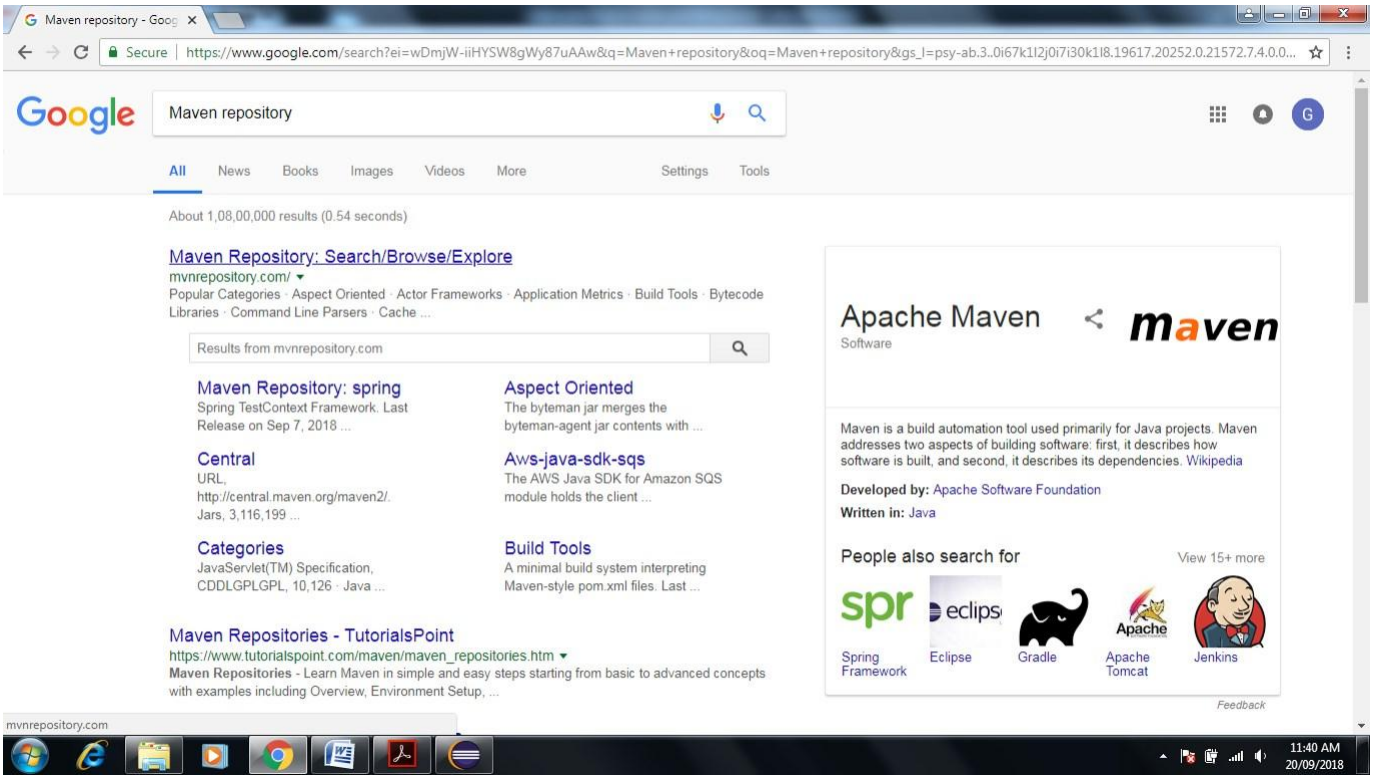
```
<modelVersion>4.0.0</modelVersion>
<groupId>MavenTestReportDemo</groupId>
<artifactId>MavenTestProject</artifactId>
<version>0.0.1-SNAPSHOT</version>
```



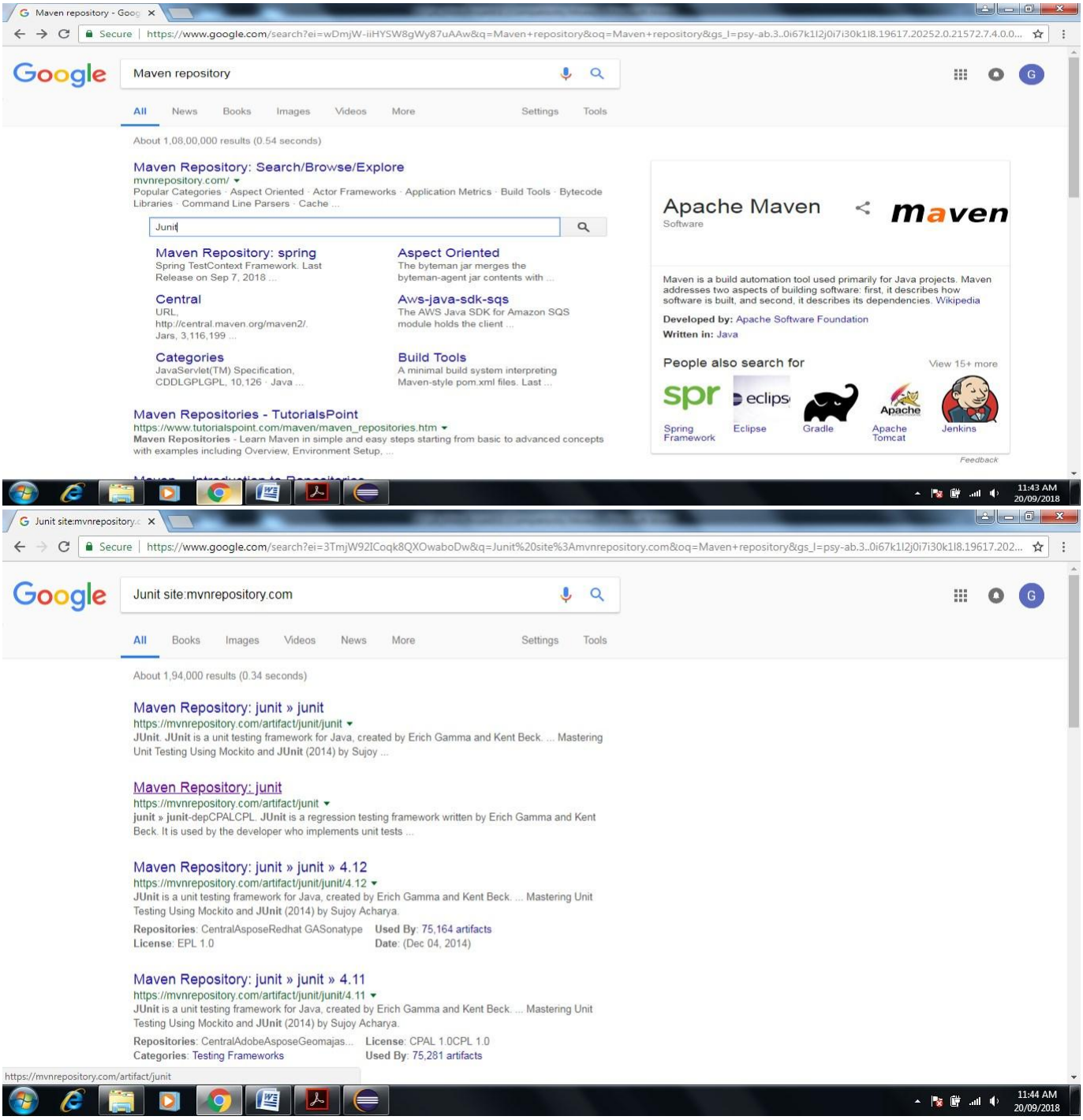
43. We add dependencies to pom.xml of Junit and Selenium



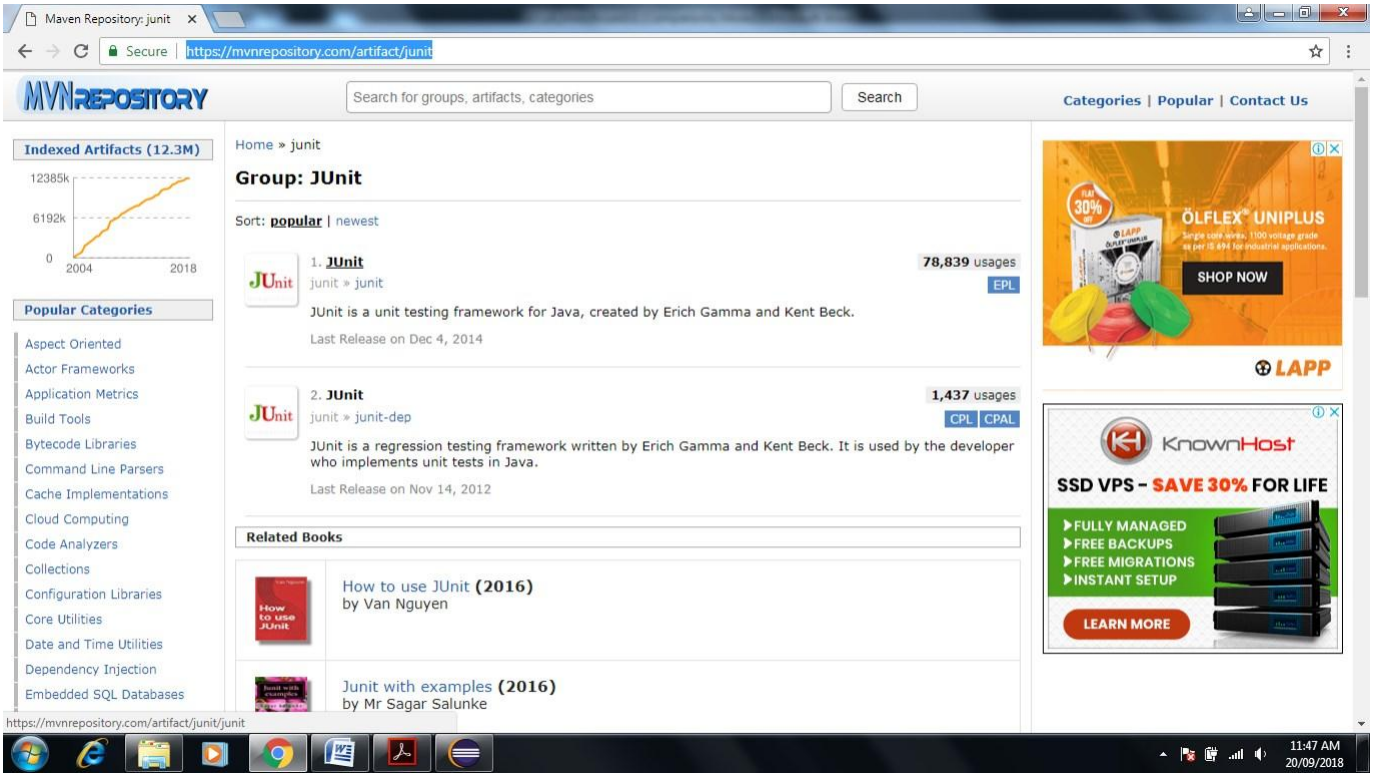
44. To add dependency→ Go to Google.com->Enter Maven repository-> in Search box on Site Enter Junit



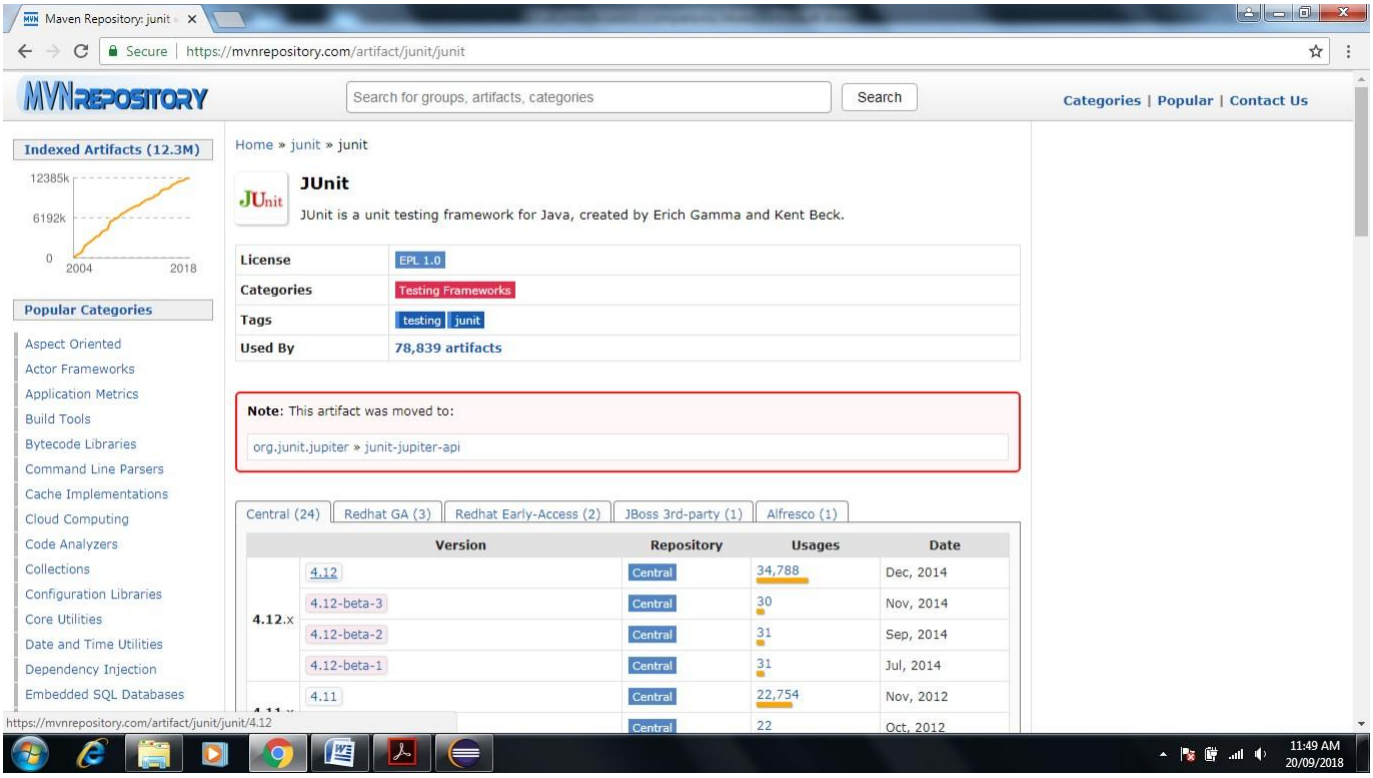
45. after Enter keyword Junit inside Seach box then Enter->it shown another Site Maven Repository for Junit Select that site.



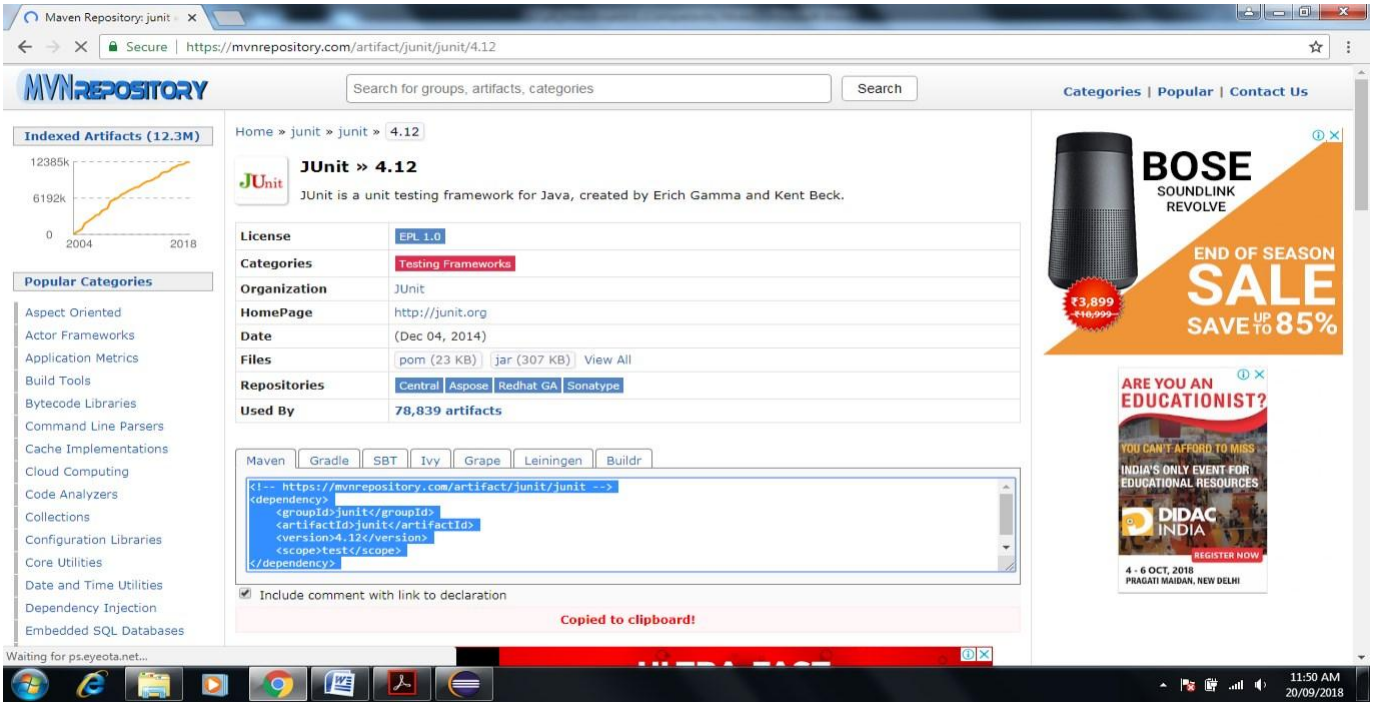
46. Click on Maven Repository-JUnit it open another site-(https://mvnrepository.com/artifact/junit)



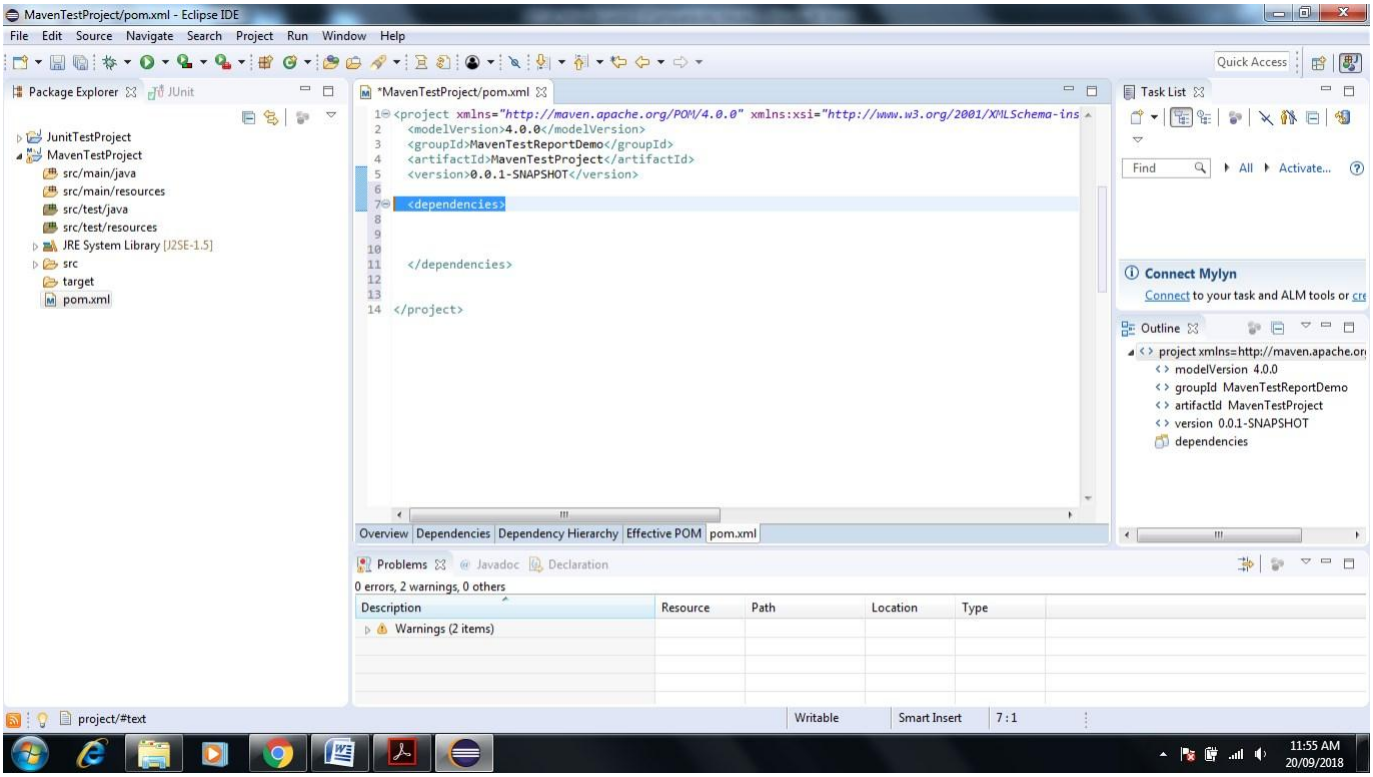
47. Click on JUnit-> Open and click on latest version as shown below (here 4.12x)



48. Copy above dependency to paste in pom.xml in Maven in Eclipse

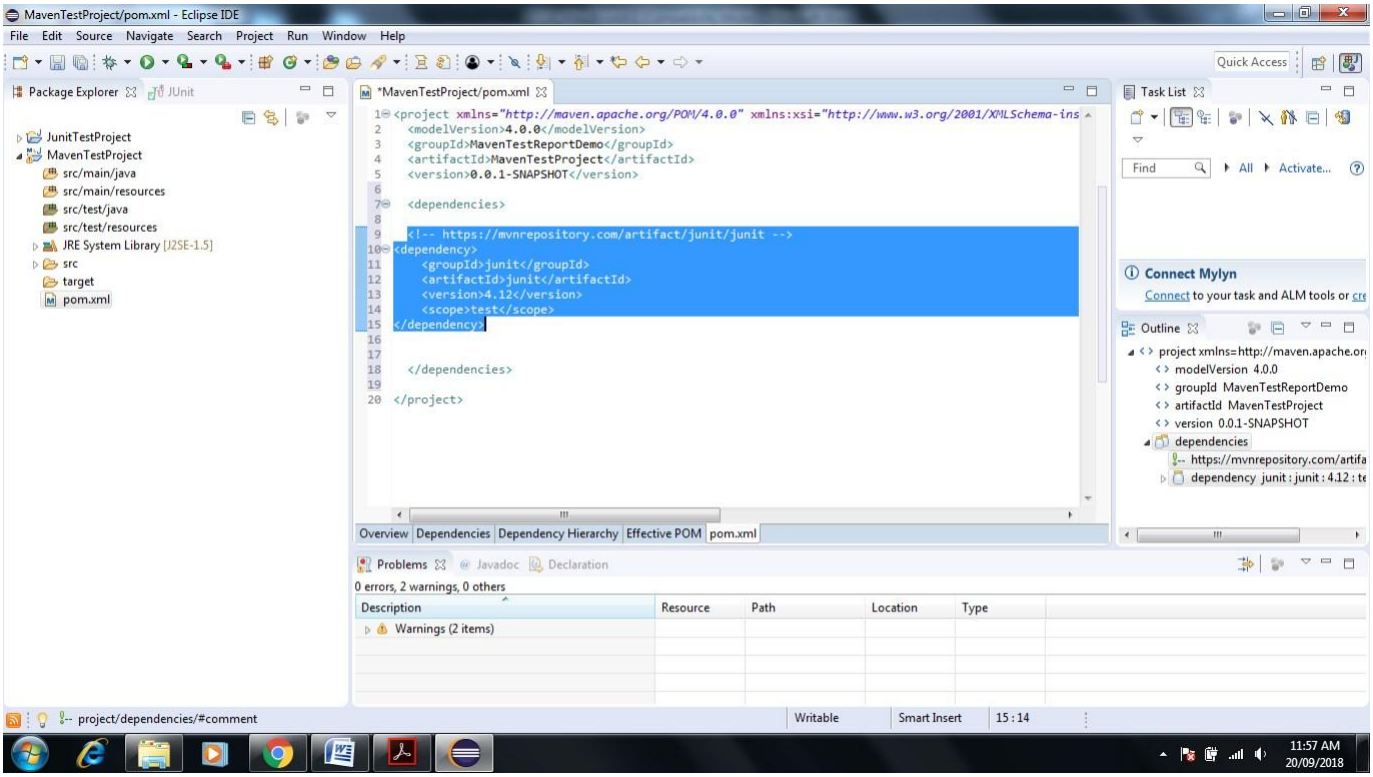


49. Add <dependencies> tag before pasting as shown below

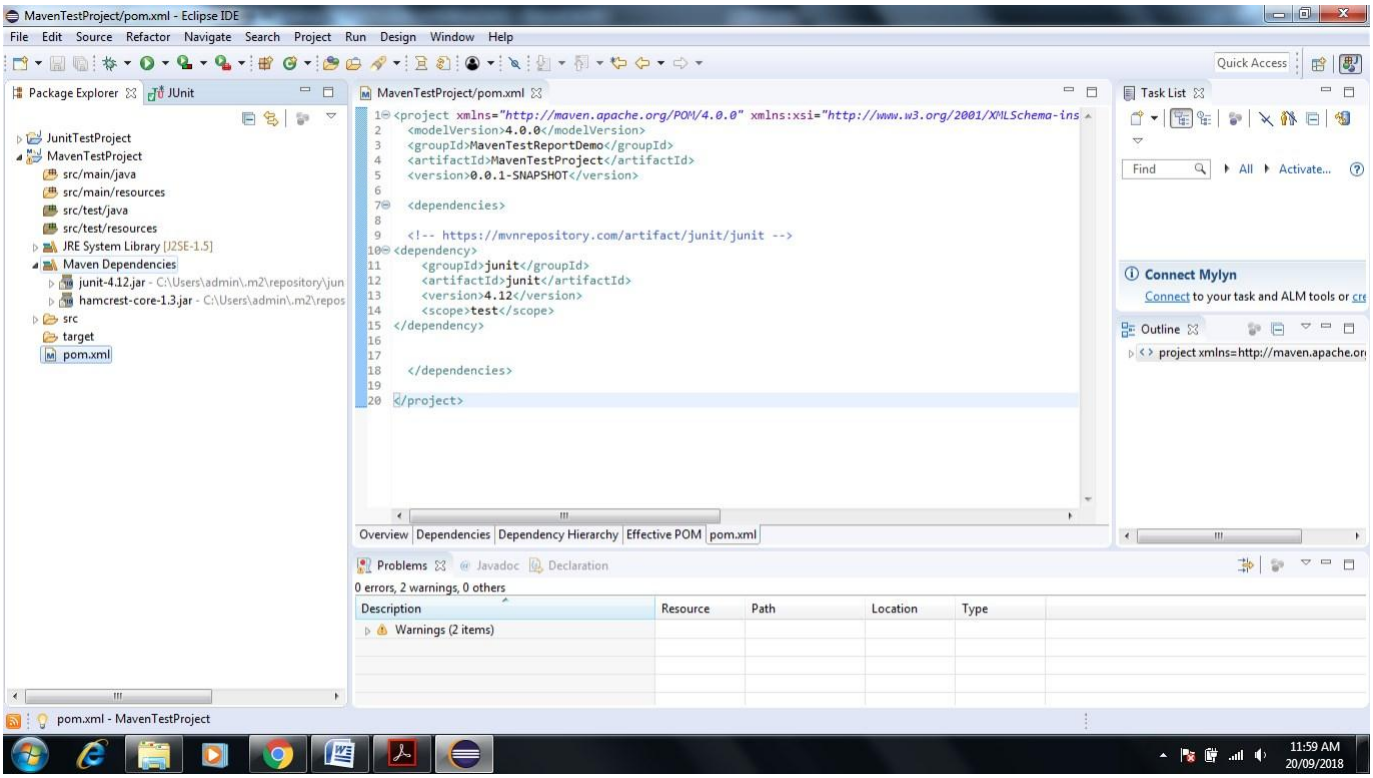


50. Now Paste the above code in between <dependencies> tag then save pom.xml file

<!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.12</version>
<scope>test</scope>
</dependency>

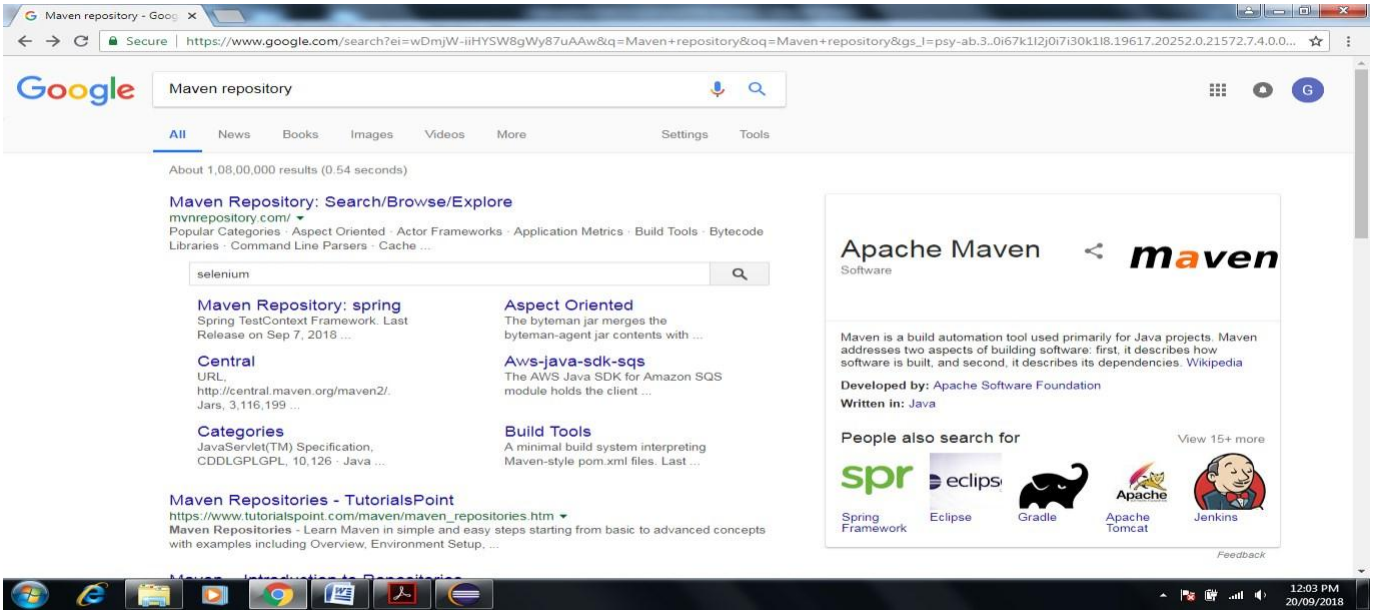


51. Now it It gets reflected in Maven by adding Junit jars

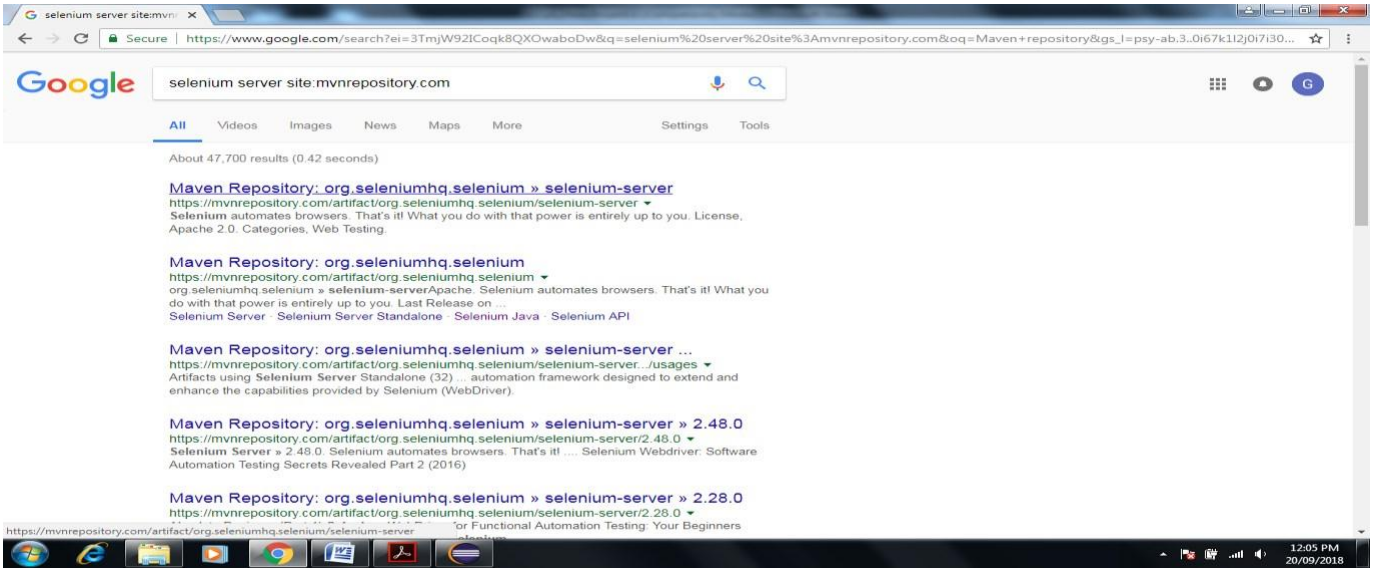


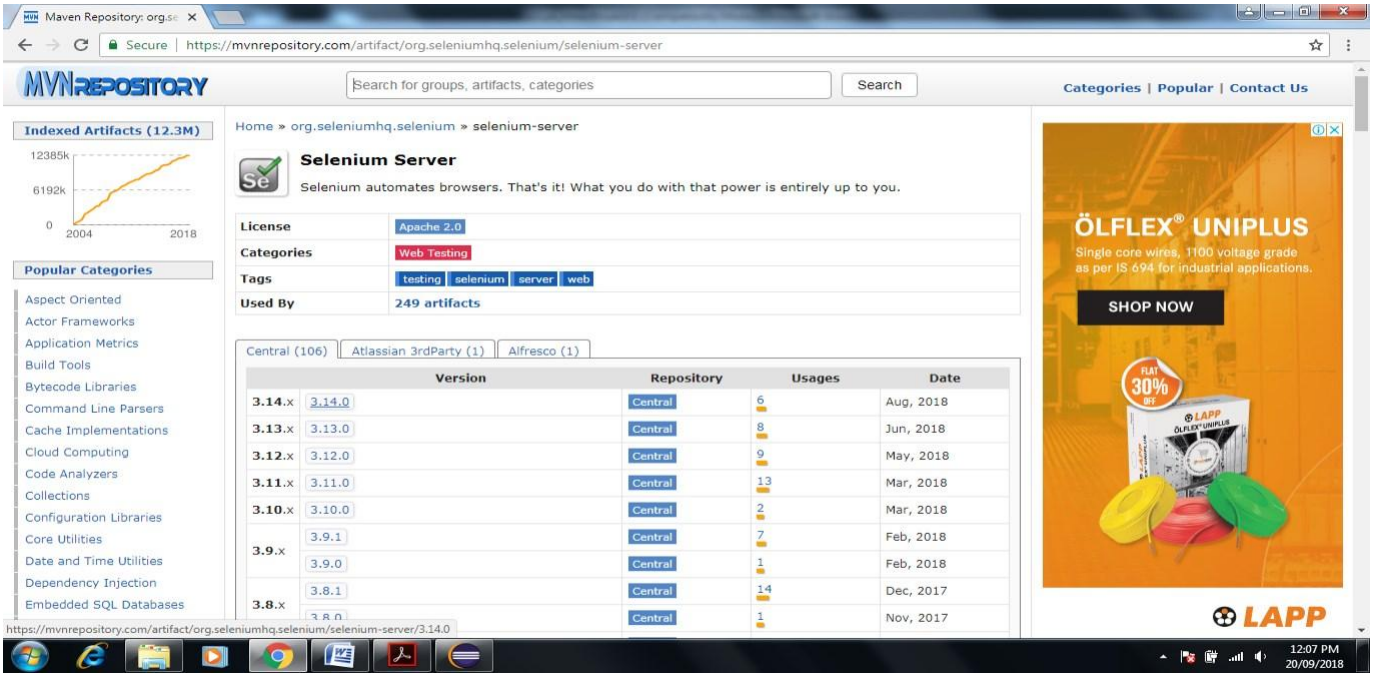
52. Same process can be repeated for Selenium server

Go to Google-> Enter Maven Repository->Enter Selenium Server in Search box->Enter

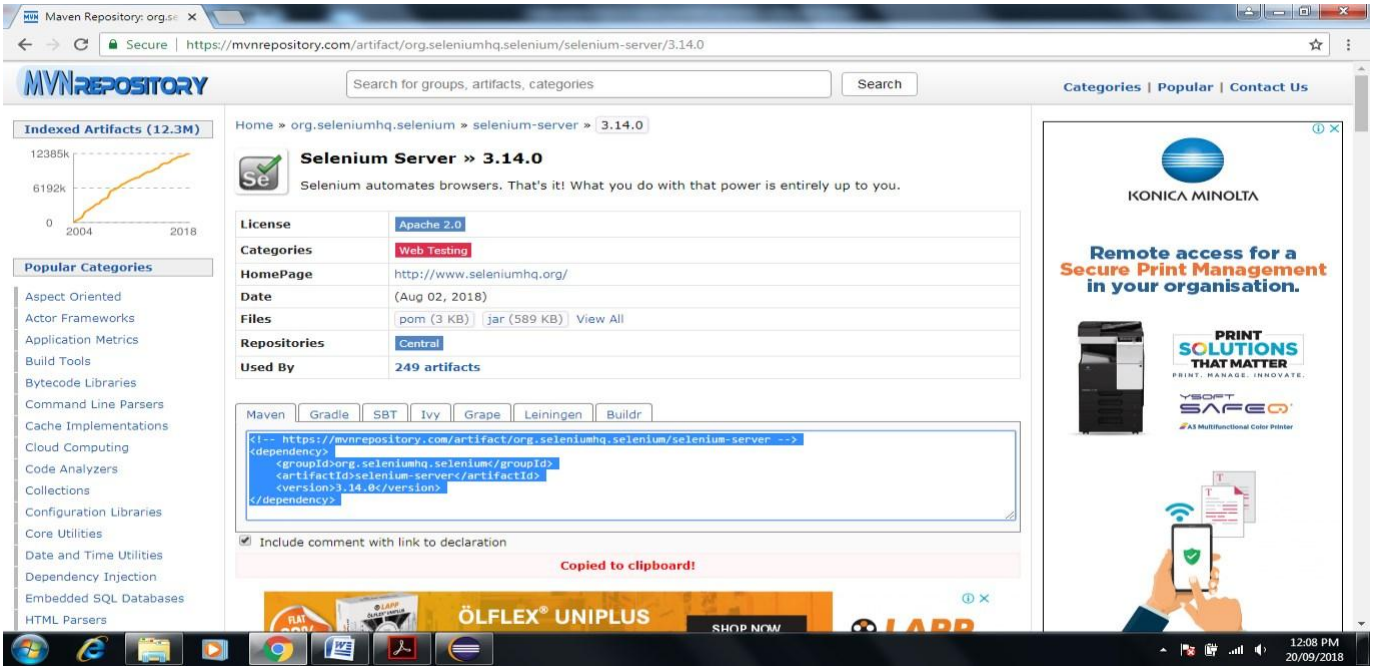


53. Click on First Link of Website-> Click on latest version

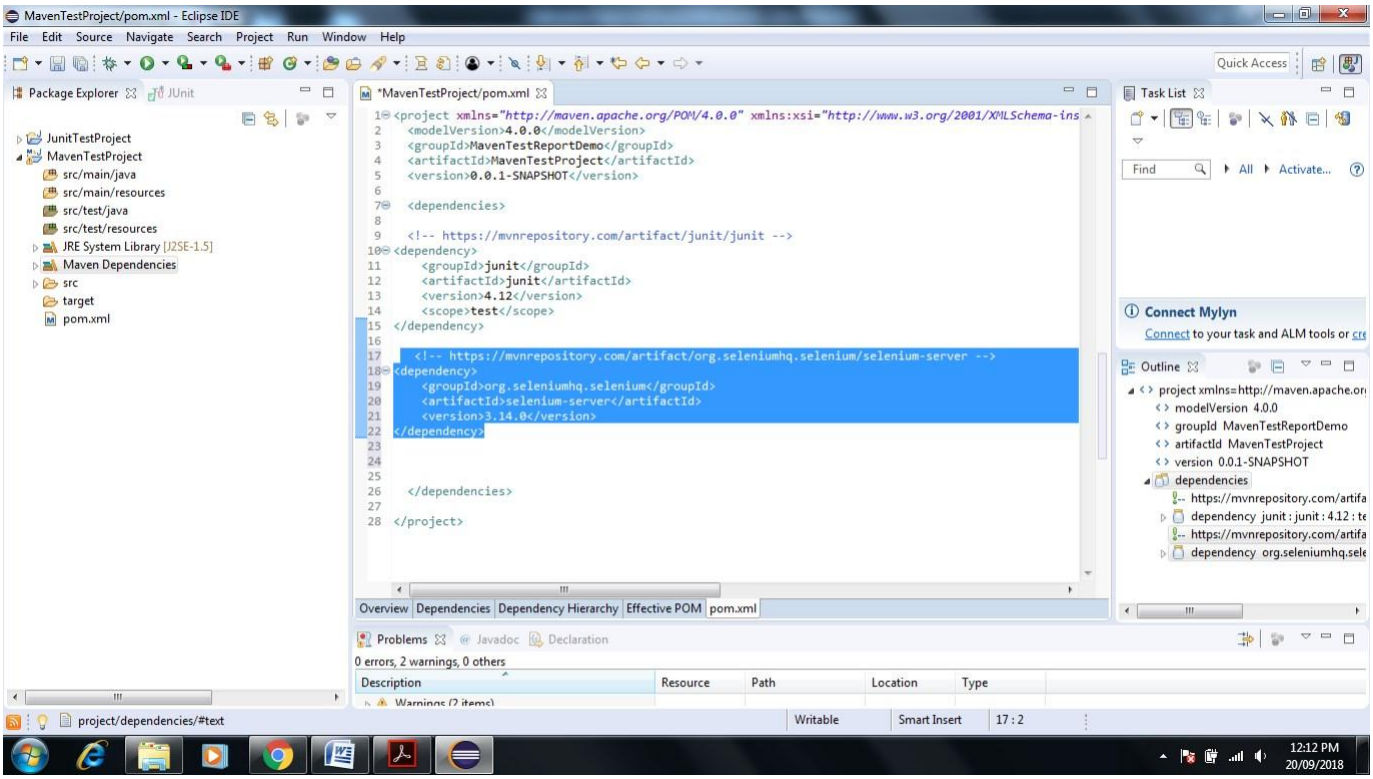




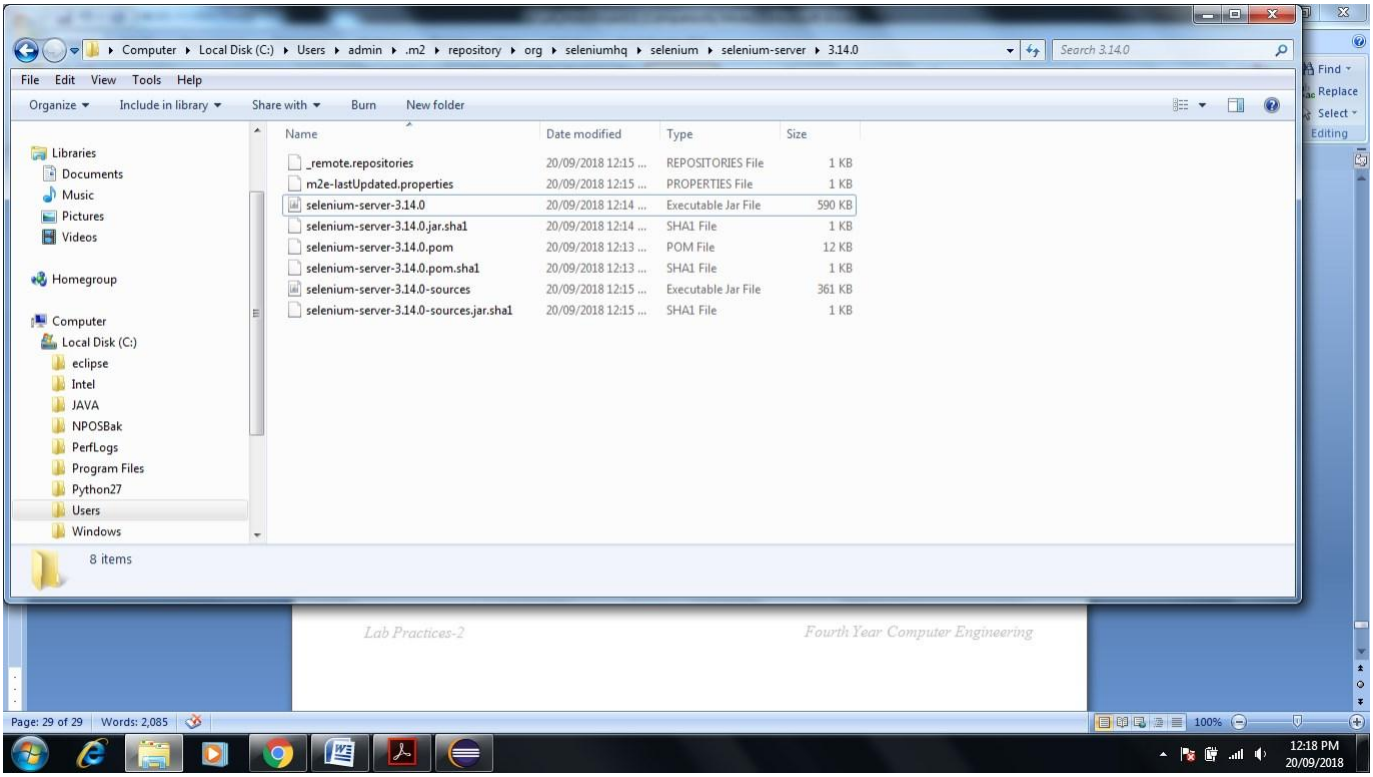
54. Copy Code in Maven Tab



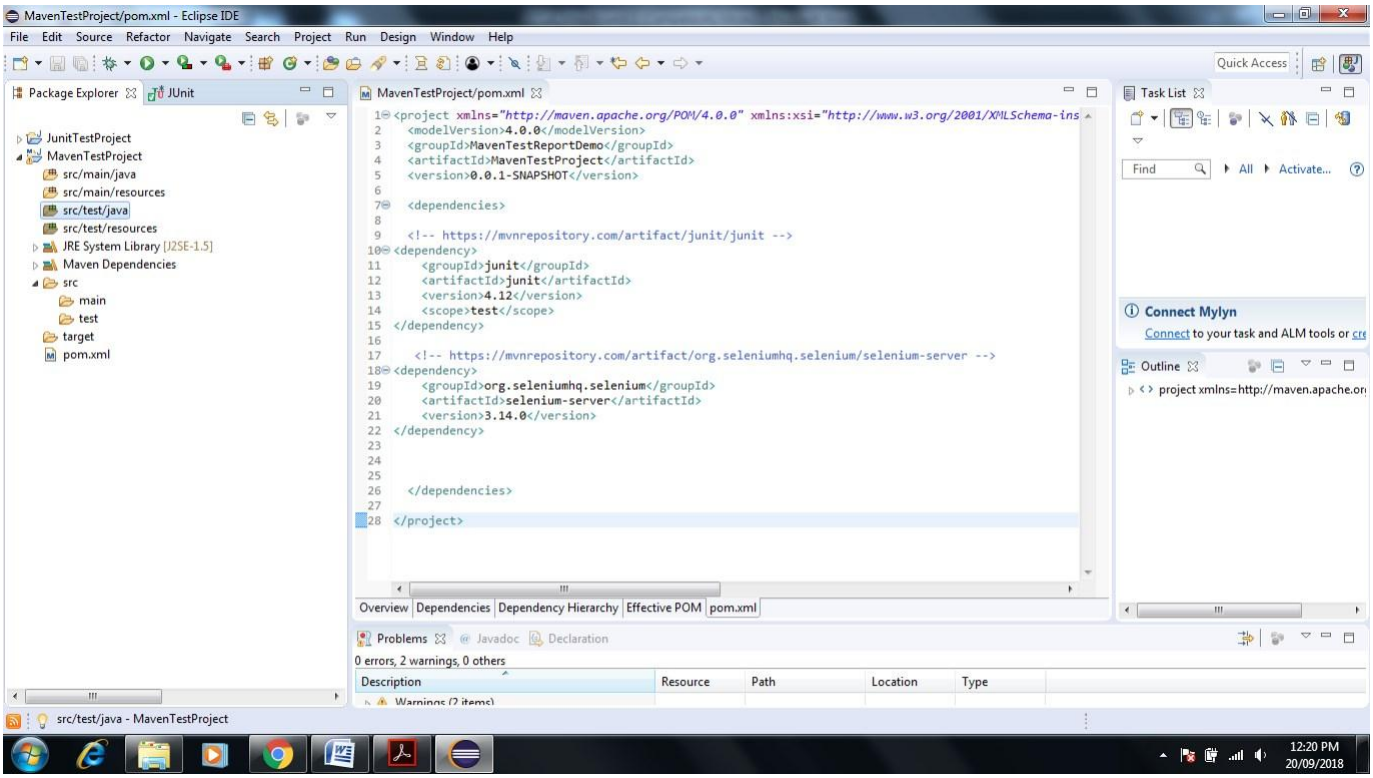
55. Paste in pom.xml file in between <dependencies> tag



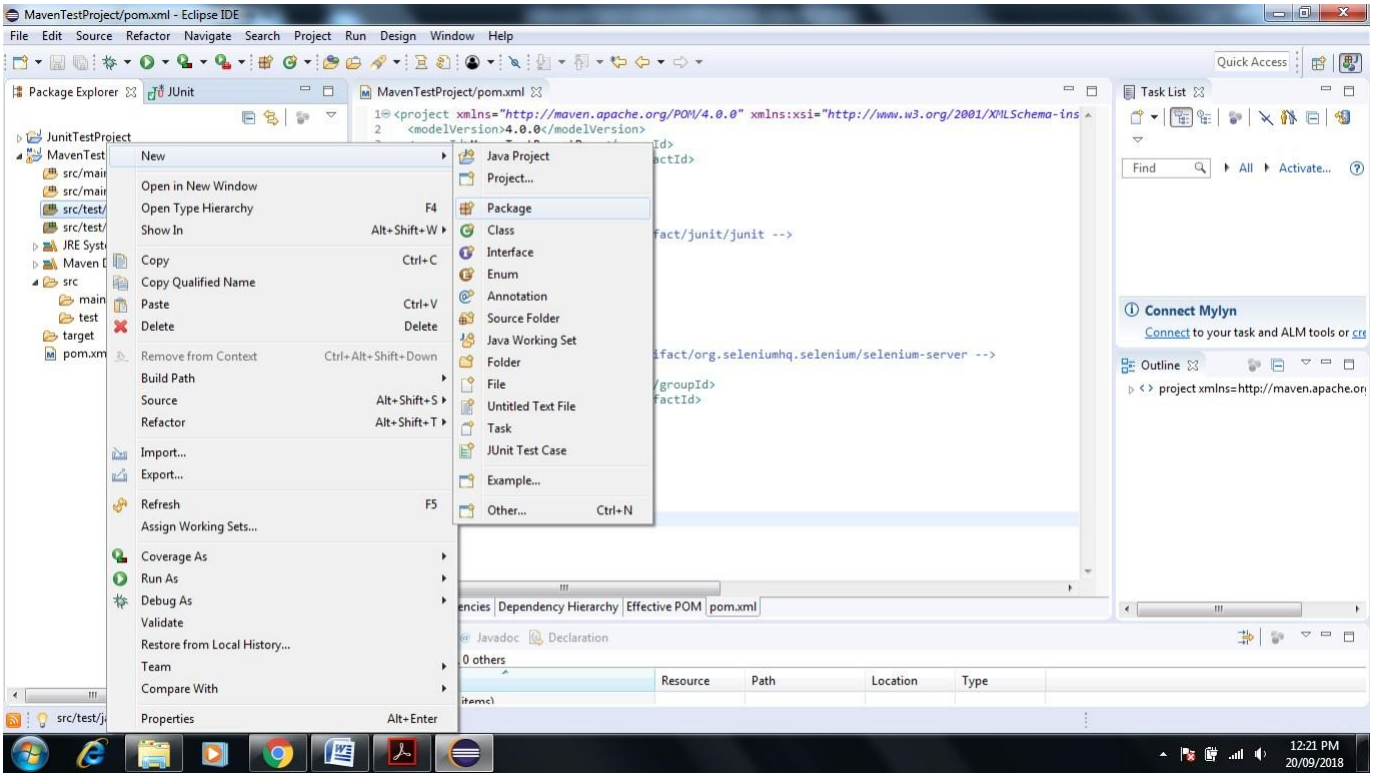
56. Now Go to C:\Users\admin\.m2\repository\org\seleniumhq\selenium\selenium-server\3.14.0 Check the latest selenium server version.



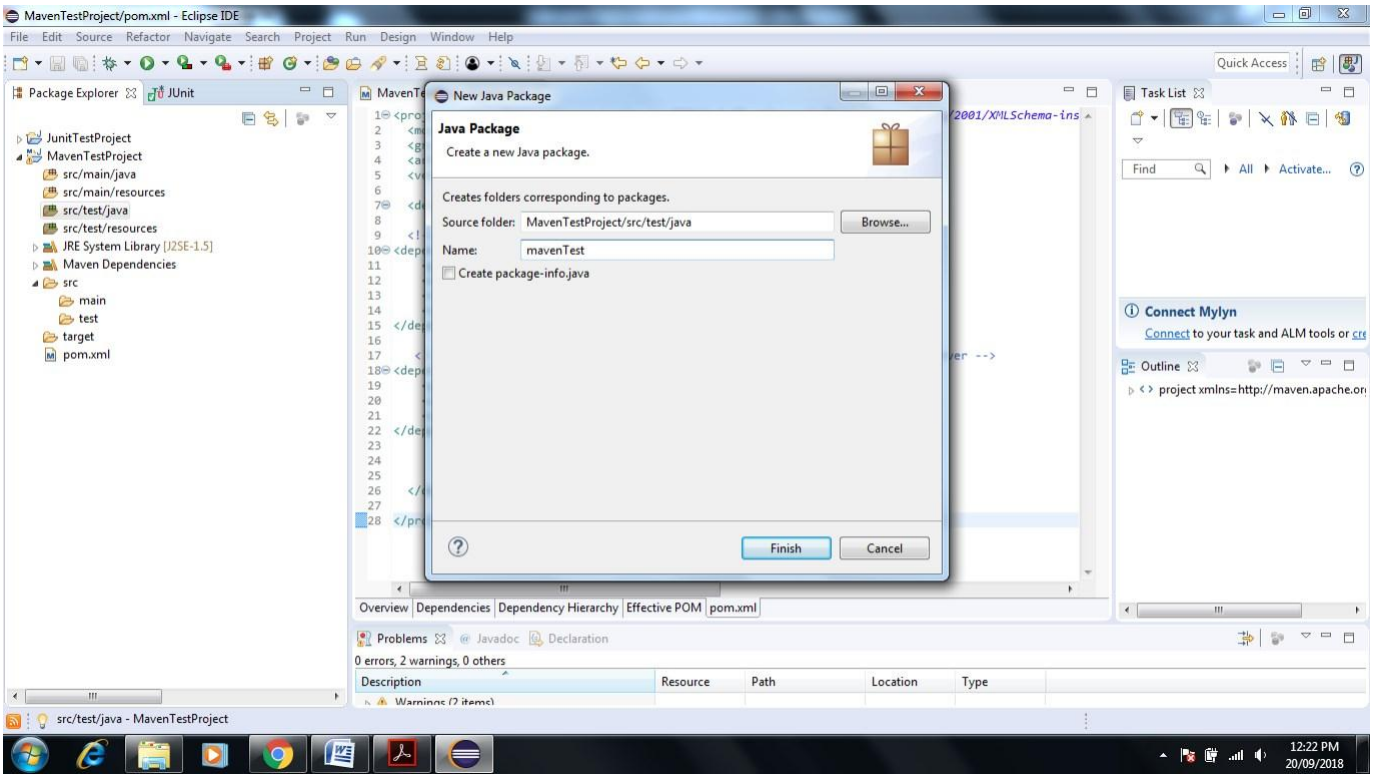
57. Now go to Eclipse -> Click on Maven Test Project->Right Click on src/test/java



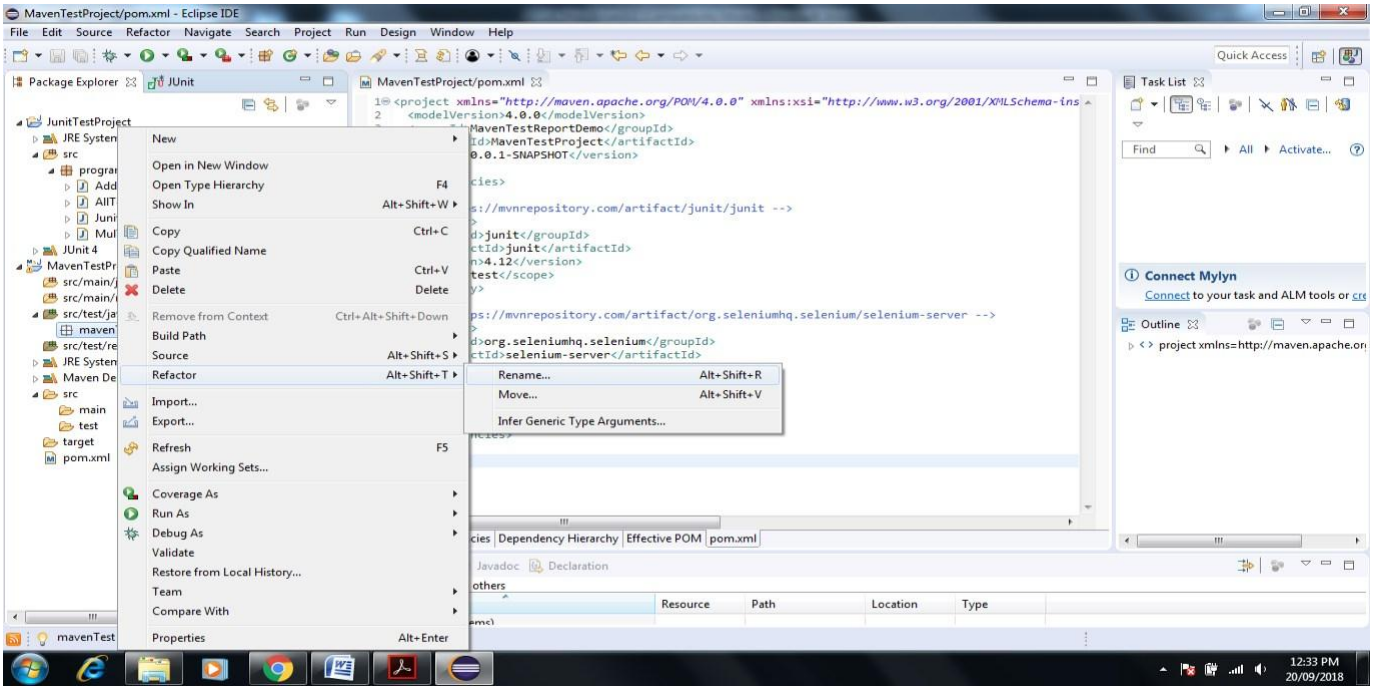
58. Click New->Package



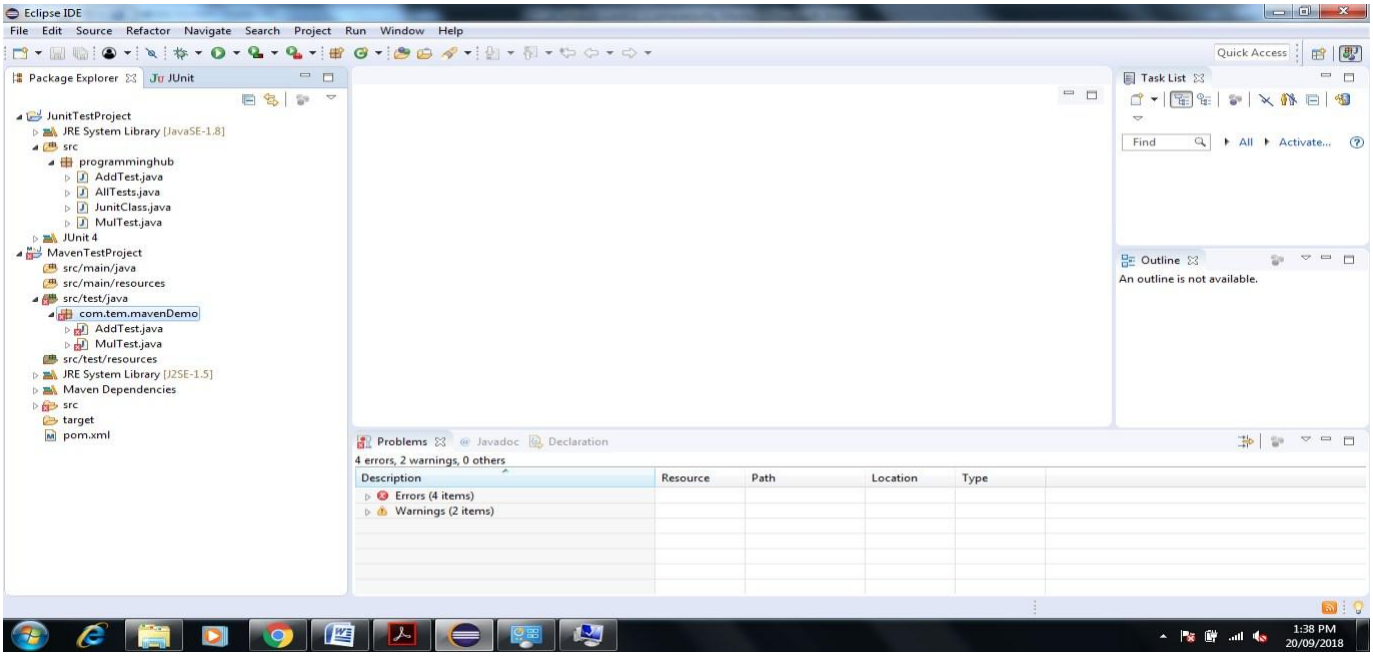
59. Give name to package mavenTest



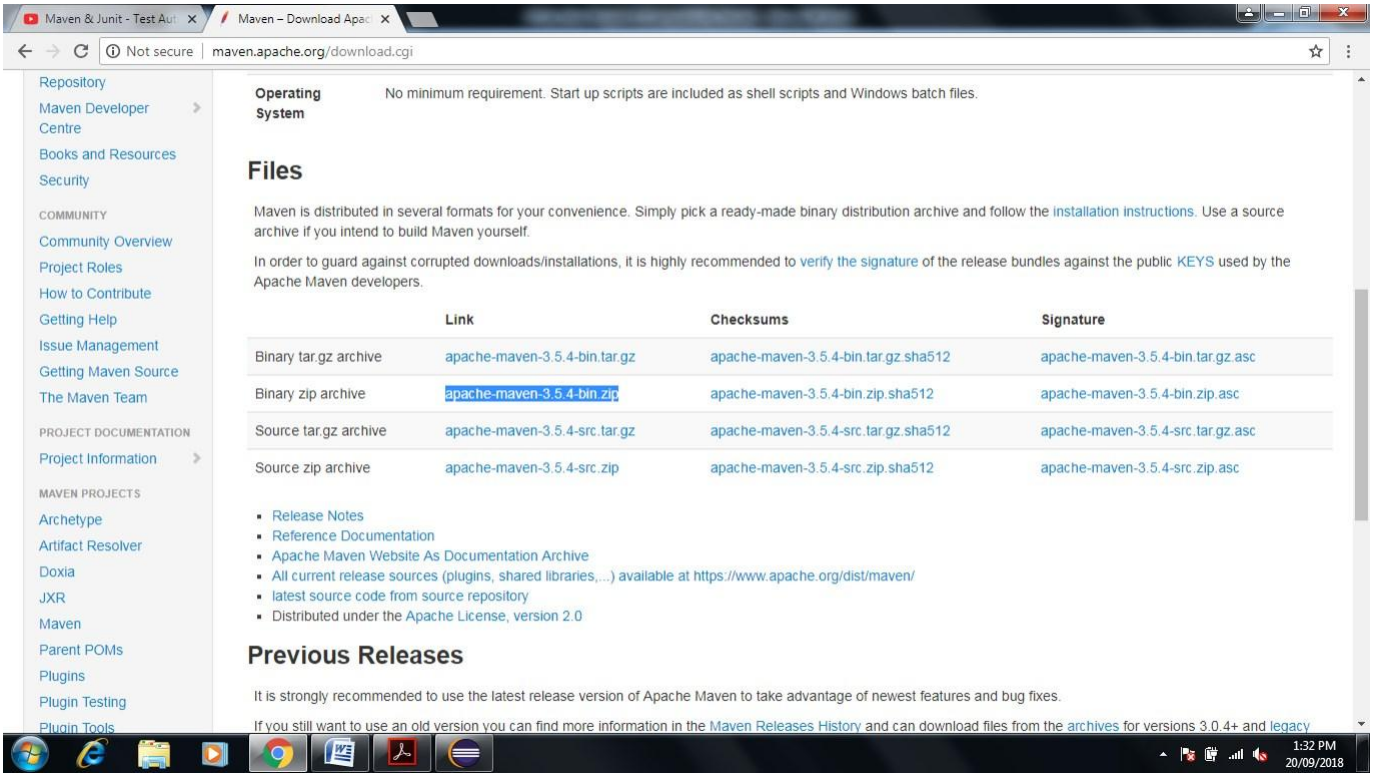
60. It shows the manvenTest Package under src/test/java folder now rename same by right click on mavenTest Click on Refactor->Rename->give another name com.tem.mavenDemo->Click on ok



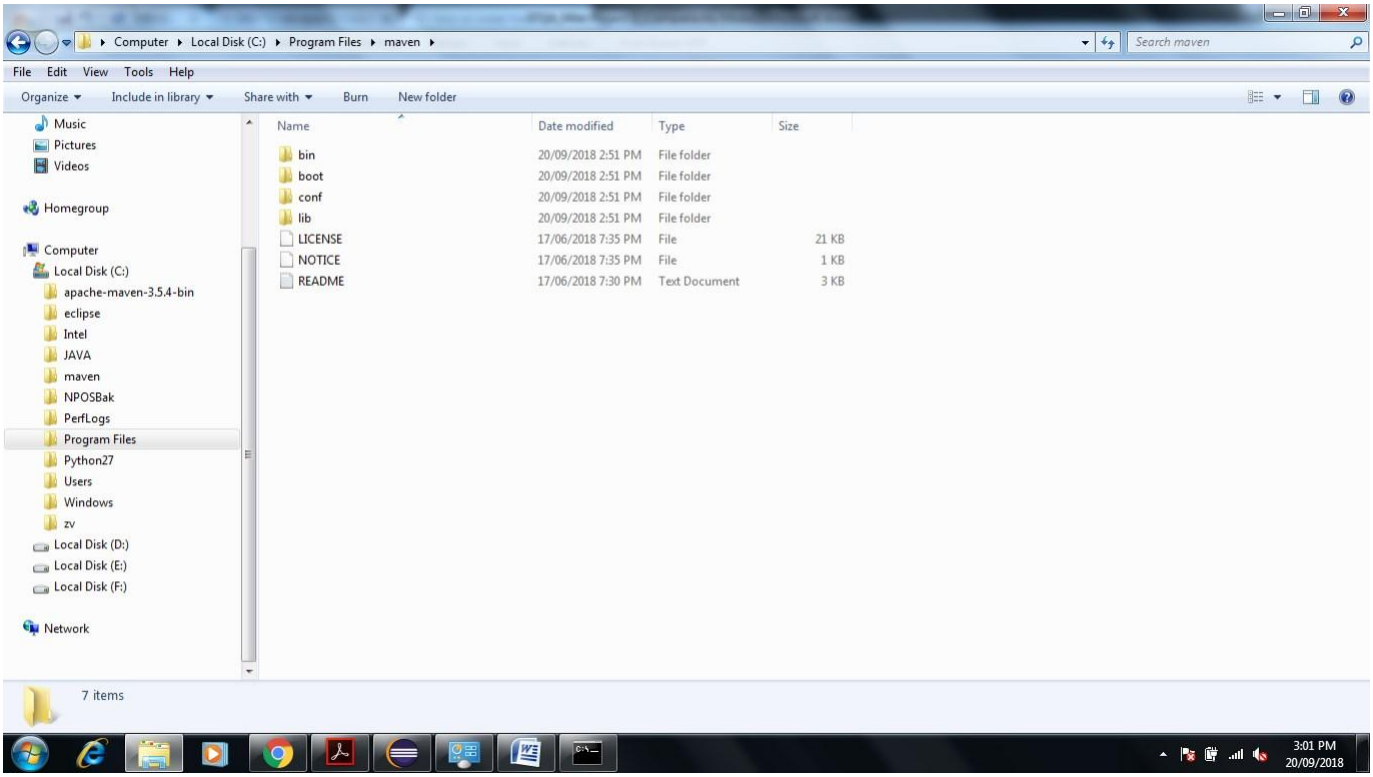
61. rename as com.tem.mavenDemo



62. Download Apache Maven Select that binary apache-maven-3.5.4-bin



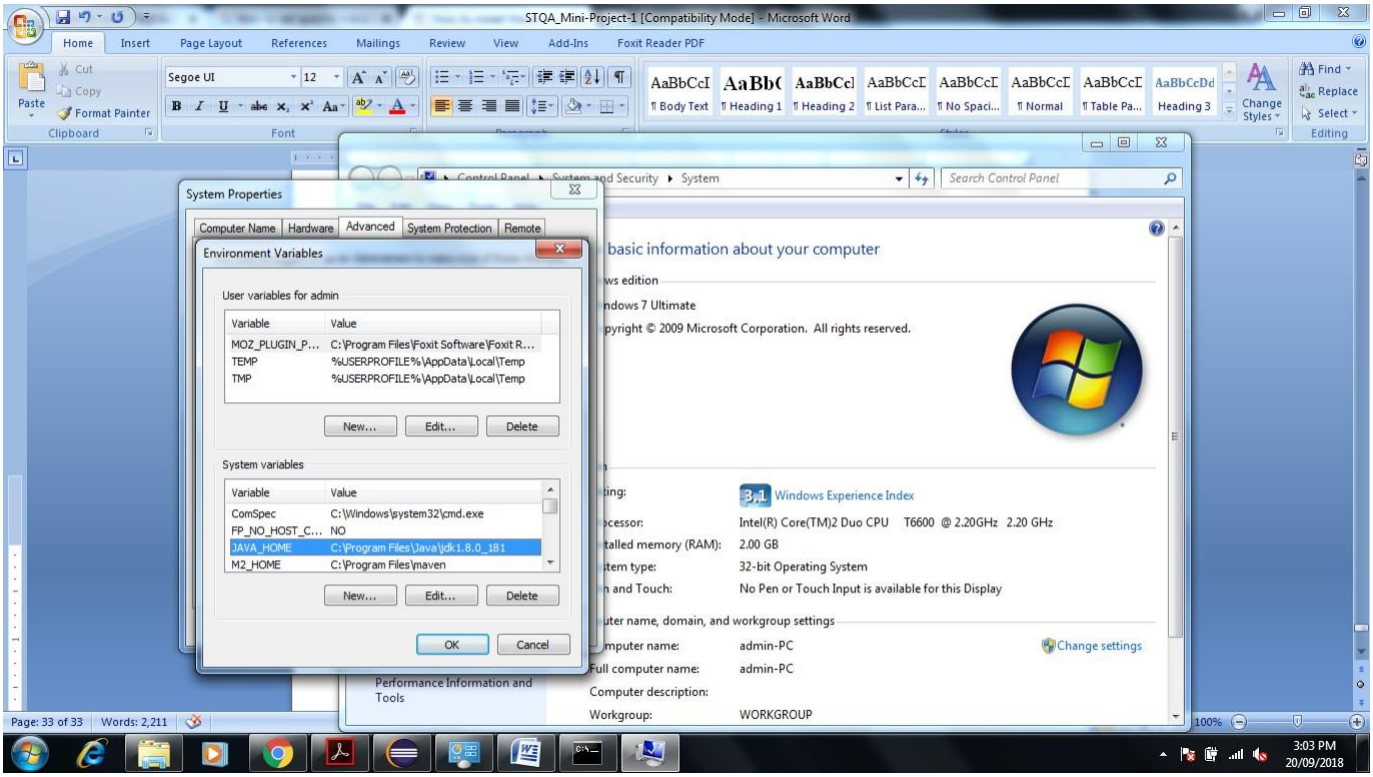
63. after Download->go to Program File->create one folder give name maven-> now extract the downloaded file in maven folder



64. Environment Setup Very Important Steps to Generate Report

1. **JDK and JAVA_HOME**

Make sure JDK is installed, and "**JAVA_HOME**" variable is added as Windows environment variable. Our JDK installed in Program File ->JAVA->JDK 1.8.0



2. Set Path of Add **M2_HOME** and **MAVEN_HOME**
   Create new system variable name M2_HOME and **MAVEN_HOME separately set variable value→ C:\Program Files\maven**

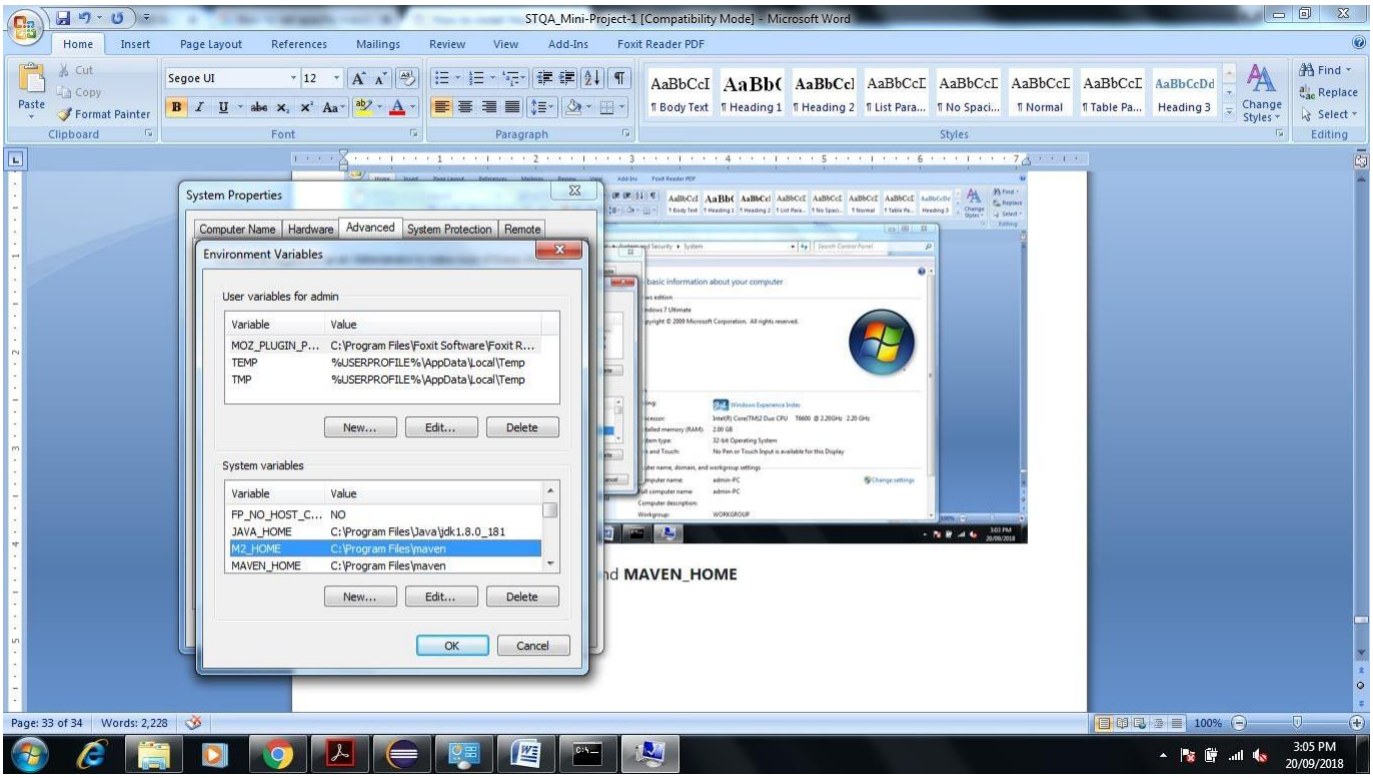


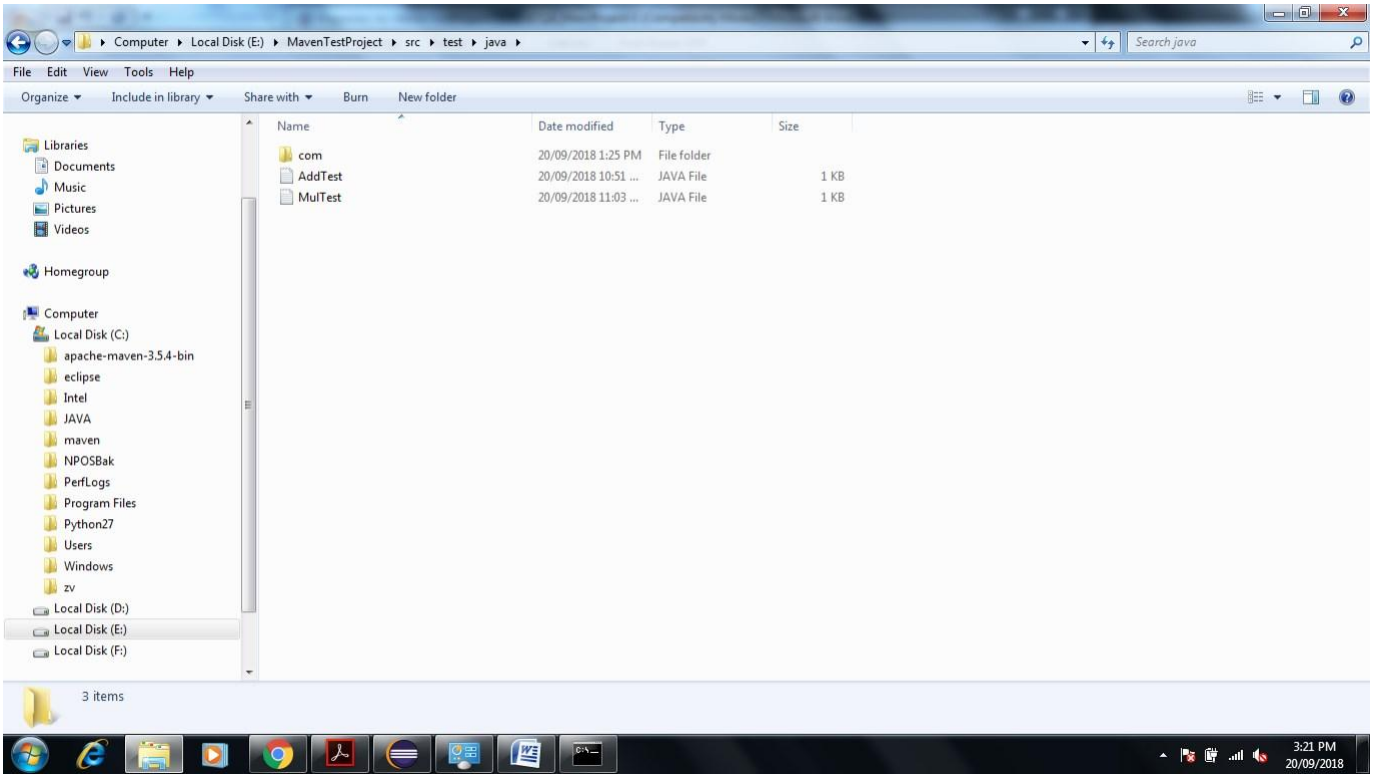
Figure Shows the Path of M2_HOME & MAVEN_HOME same.

3. **Update PATH Variable as per folowing**

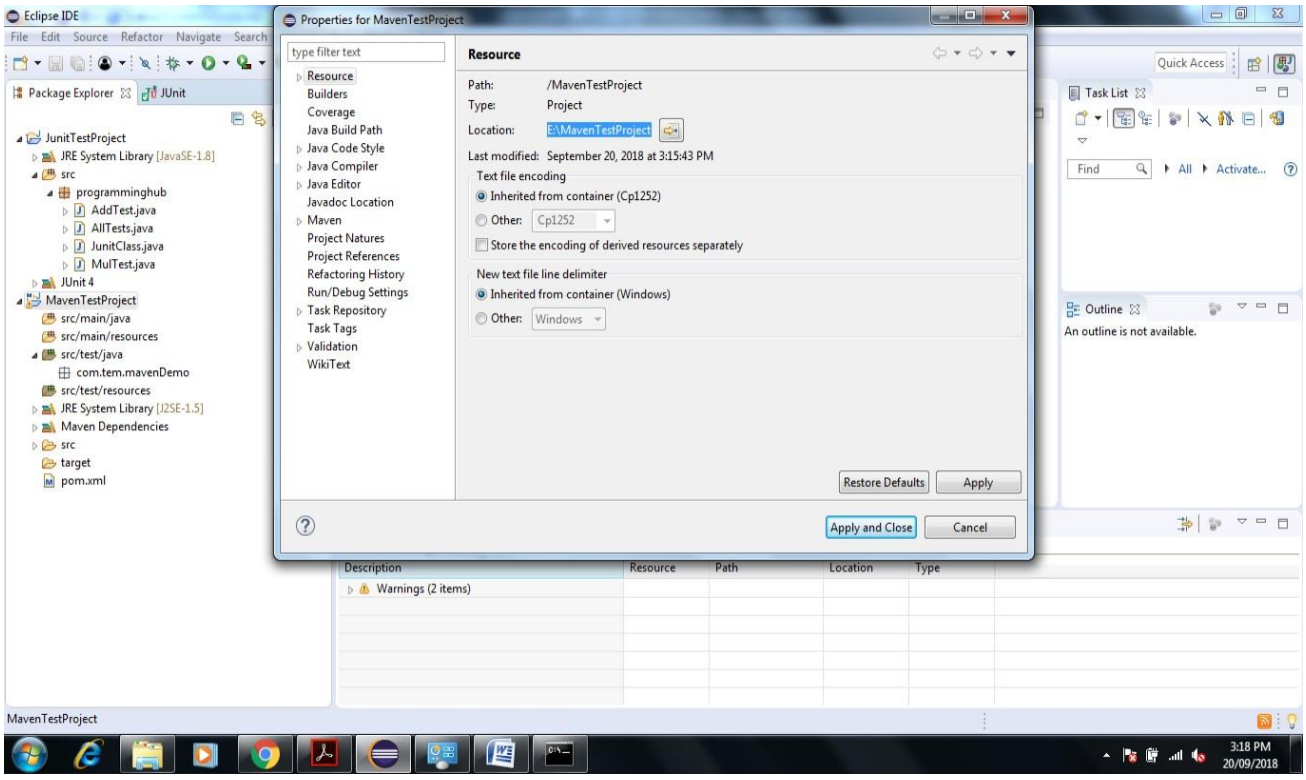
C:\Program Files\maven%MAVEN_HOME%\bin;%M2_HOME%\bin;

4. Verification

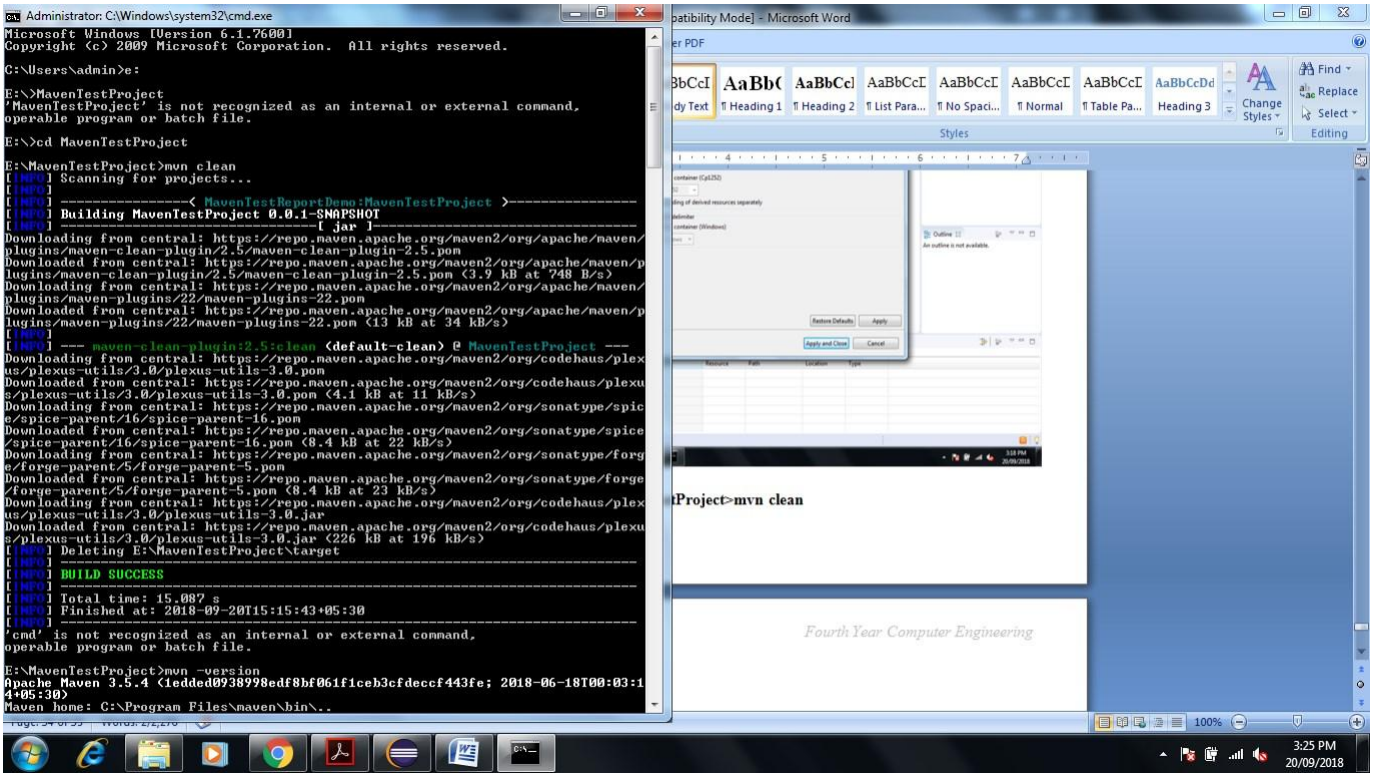
Now copy Previous Created JUnit Test case java file Add Test and Mul Test Paste Externally in E:\MavenTestProject\src\test\java



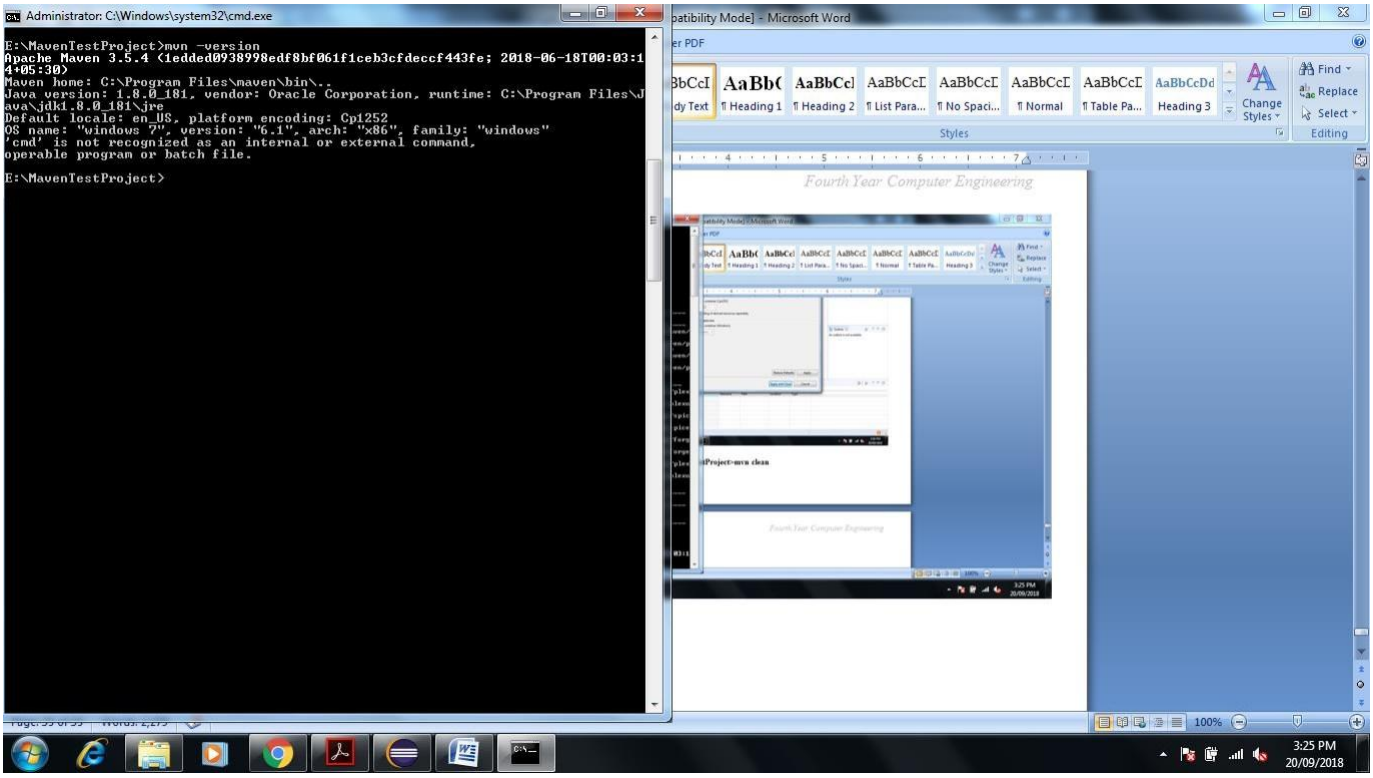
Now Open Eclipse Right Click on MavenTestProject->Properties→Resources→Copy Path of Project Folder



Now go to command prompt→E:\MavenTestProject>mvn clean



Enter E:\MavenTestProject>mvn –version



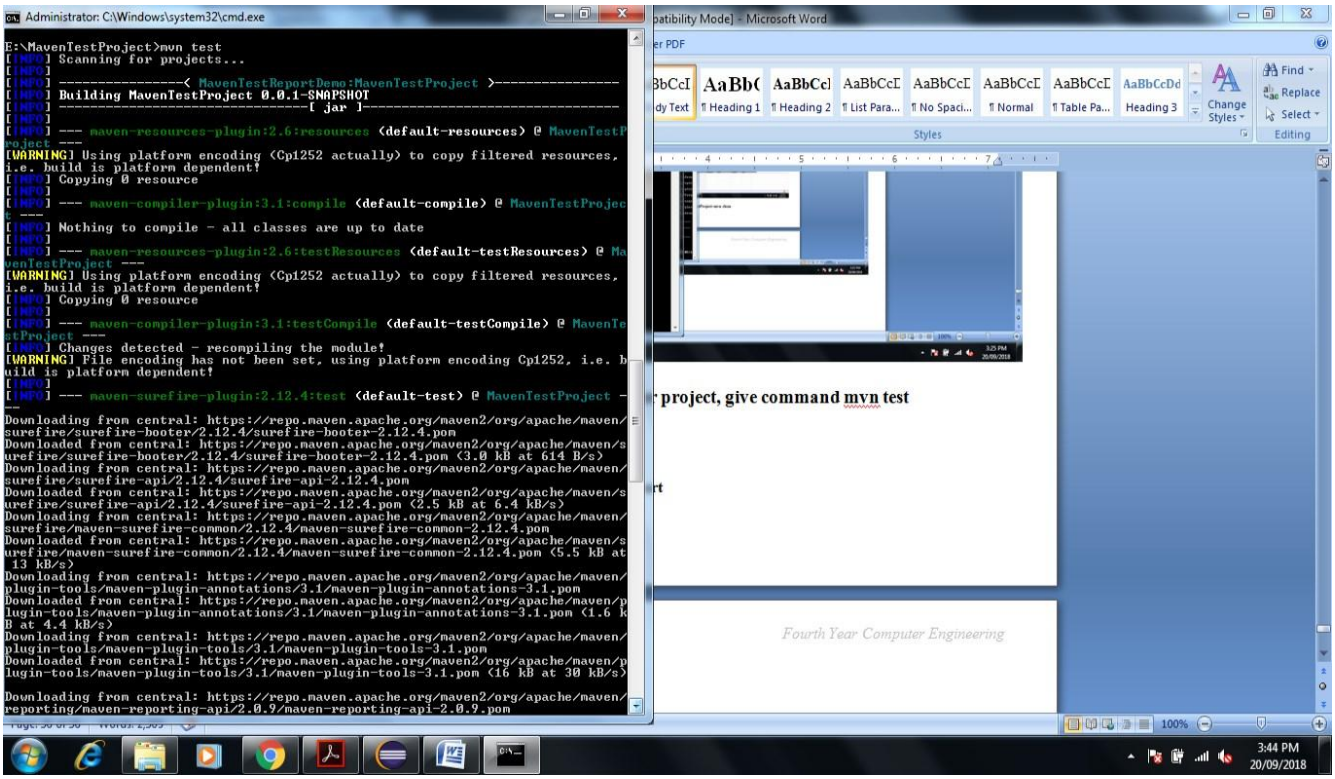
To run test suite or all test cases under project, give command mvn test

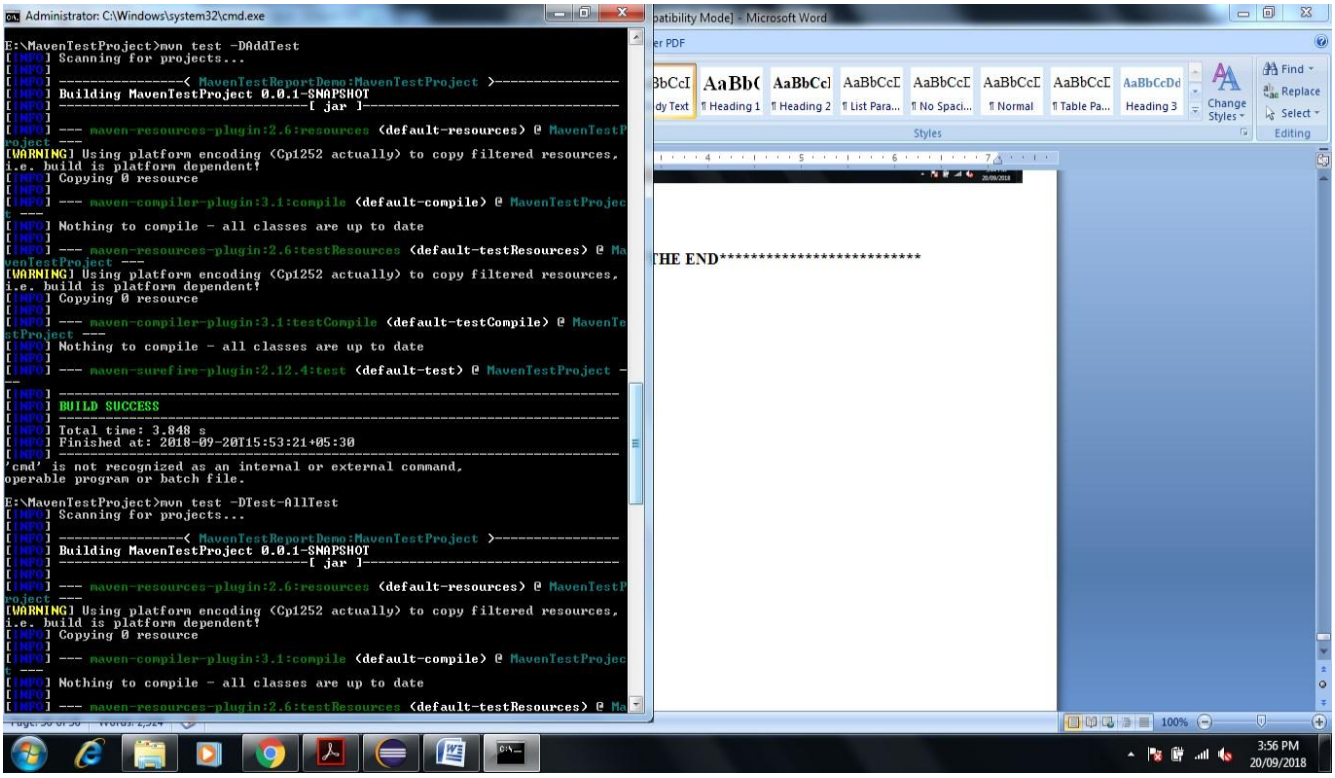
Enter E:\MavenTestProject>mvn test

This Command is Used to See the Test Report

You can even run individual test cases. Give command mvn test –Dtesttestcasename

Eg. mvn test –Dtest-AllTest





1.8.8 Conclusion

In this way using JUnit and Maven Automation tool we are Perform Unit Testing and Prepare Test Report of same.

1.8.9 Assignment Question

1. Write any Five Tool for White Box and Black Box Testing Purpose.

STQA Mini Project No. 2

2.1 Title

Create a small web-based application by selecting relevant system environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

2.2 Problem Definition:

Perform Web testing and identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

2.3 Prerequisite:

Knowledge of Core Java

2.4 Software Requirements:

Eclipse photon R latest Version, JAVA 1.8, selenium-server-standalone-3.13.0 Chromedriver.exe

2.5 Hardware Requirement:

PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

2.6 Learning Objectives:

We are going to learn how Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

2.7 Outcomes:

You are able to Web Testing using Automation Tool like Selenium Web driver and IDE

2.8 TheoryConcepts:

2.8.1 What is Selenium?

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.

Selenium is a suite of software tools to automate Web Browsers.

• It is an Open source suite of tools mainly used for Functional and Regression Test Automation.

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.

It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred as Selenium Testing.

• **Selenium supports various Operating environments.**

- ✓ MS Windows
- ✓ Linux
- ✓ Macintosh etc…

• **Selenium supports various Browsers.**

- ✓ Mozilla Firefox
- ✓ IE
- ✓ Google Chrome
- ✓ Safari
- ✓ Opera etc…

Note: **Selenium IDE supports Mozilla Firefox only.**

• **Selenium supports various programming environments to write programs (Test scripts)**

- ✓ Java
- ✓ C#
- ✓ Python
- ✓ Perl
- ✓ Ruby
- ✓ PHP

2.8.2 History of the Selenium Project

Selenium first came to life in 2004.

• In 2006, Selenium WebDriver was launched at Google.

• In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC in order to form more powerful tool called Selenium 2.0

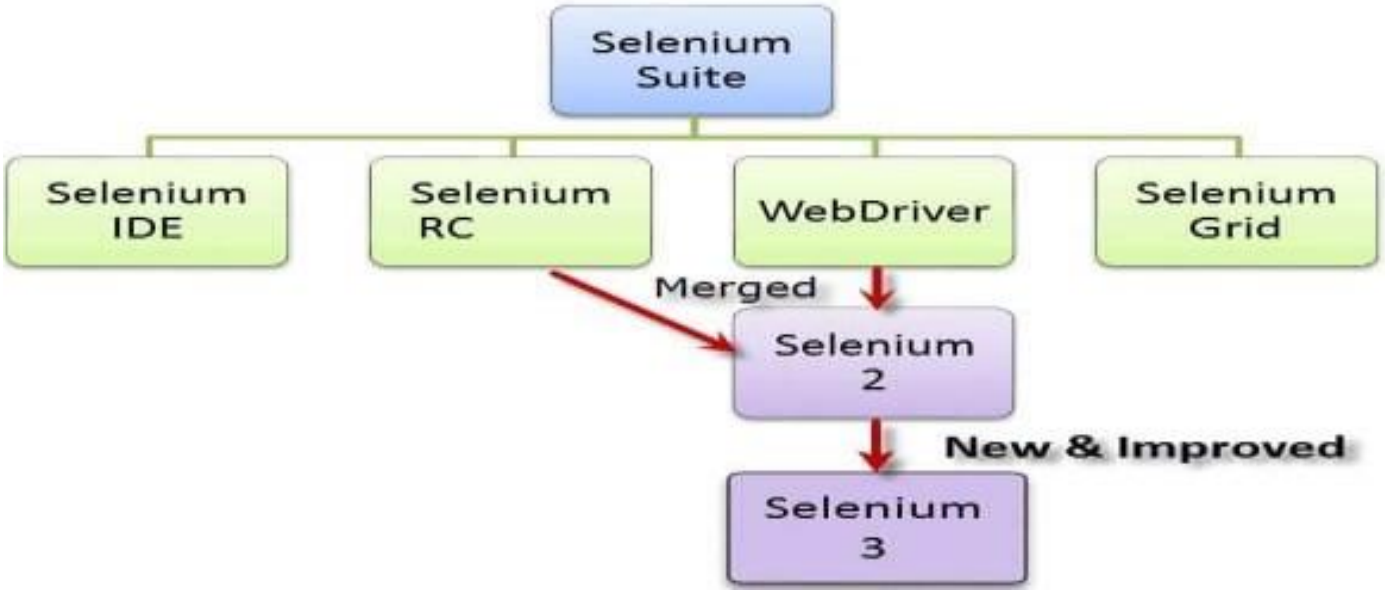
- ✓ **Selenium 1**
  (Selenium IDE + Selenium RC + Selenium Grid)
- ✓ **Selenium 2**
  (Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)

2.8.3 Selenium's Tools Suite

Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization.

It has four components.

- **Selenium Integrated Development Environment (IDE)**
- **Selenium Remote Control (RC)**
- **WebDriver**
- **Selenium Grid**



2.8.4 Brief Introduction Selenium IDE

It is a Firefox browser plug in, used to create and execute Test cases.

1. **Selenium IDE Features:**

• Create Test Cases, Test suites (We can Record test cases or type Test steps using element locators and Selenese commands)

• Edit Test Cases

• Execute Test cases, Test suites

• Debug Test Cases.

• Enhance Test Cases

• Export Test cases to other formats (java, ruby etc…)

Note: **selenium IDE Test case default format is .html**

2. Drawbacks of Selenium IDE

• It supports Mozilla Firefox browser only.

• It doesn't support Programming logic/features to enhance Test cases.

• It doesn't support Data Driven Testing.

• It is not suitable for complex test case design.

• No centralized maintenance of Objects/Elements

3. Selenium RC (* Out dated) -Currently, Selenium RC is still being developed but only in maintenance mode.

4. Selenium WebDriver

- ✓ It is a Programming interface to create and execute Test cases.

  Selenium IDE has IDE but doesn't have Programming interface
- ✓ Selenium WebDriver has Programming interface but doesn't have IDE
- ✓ It communicates Directly to the browser.
- ✓ No need of Separate Server such as RC Server
- ✓ UFT/QTP has both IDE as well as Programming interface
- ✓ Faster Execution than IDE & RC

• **Selenium WebDriver supports various programming environments to write programs.**

- ✓ Java,
- ✓ C#
- ✓ Perl
- ✓ Python
- ✓ Ruby
- ✓ PHP

• Using Element/Object locators/properties and Webdriver Methods we can create and execute Test cases.

• Selenium Webdriver supports various browsers to create and execute test case/test script/test

Note: **Browser driver varies from one browser to another.**

• **Selenium WebDriver supports various operating environments**

- ✓ MS Windows
- ✓ Linux Macintosh etc…

Drawback of Selenium WebDriver

• It doesn't generate detailed Test Reports.

• No centralized maintenance of Object/elements

• It require Programming Knowledge

• cannot support the readily new browser

• Installation is More Complicated than Selenium IDE

• No built-in mechanism for logging runtime message

5. Selenium Grid

• Selenium Grid is used to execute tests across multiple browsers, operating environments and machines in parallel.
• Selenium Grid 2 supports Selenium RC Tests as well as Selenium WebDriver Tests.
i) Selenium WebDriver to create Test cases using element locators and Webdriver methods.
ii) Java Programming to enhance test cases.

iii) TestNG Framework to group test cases, execute test batches and generate detailed test reports.

Features:

- Enables **simultaneous running of tests** in **multiple browsers and environments.**
- **Saves time** enormously.
- Utilizes the **hub-and-nodes** concept. The hub acts as a central source of Selenium commands to each node connected to it.

Note on Browser and Environment Support

- Because of their architectural differences, Selenium IDE, Selenium RC, and WebDriver support different sets of browsers and operating environments.

	Selenium IDE	**WebDriver**
Browser Support	Mozilla Firefox	Internet Explorer versions 6 to 11, both 32 and 64-bit Microsoft Edge version 12.10240 & above (partial support some

	Selenium IDE	WebDriver
		functionalities under development) Firefox 3.0 and above Google Chrome 12.0. and above Opera 11.5 and above Android - 2.3 and above for phones and tablets (devices & emulators) iOS 3+ for phones (devices & emulators) and 3.2+ for tablets (devices & emulators) HtmlUnit 2.9 and above
Operating System	Windows,Mac OS X, Linux	All operating systems where the browsers above can run.

- **Note:** Selenium WebDriver is termed as the successor of Selenium RC which has been deprecated & officially announced by SeleniumHQ.

2.8.5 How to Choose the Right Selenium Tool for Your Need

Tool	Why Choose?
Selenium IDE	• To learn about concepts on automated testing and Selenium, including: • Selenese commands such as type, open, clickAndWait, assert, verify, etc. • Locators such as id, name, xpath, css selector, etc. • Executing customized JavaScript code using runScript • Exporting test cases in various formats. • To create tests with little or no prior knowledge in programming. • To create simple test cases and test suites that you can export later to RC or WebDriver. • To test a web application against Firefox only.
Selenium RC	• To design a test using a more expressive language than Selenese • To run your test against different browsers (except HtmlUnit) on different operating systems. • To deploy your tests across multiple environments using Selenium Grid.

Tool	Why Choose?
	• To test your application against a new browser that supports JavaScript. • To test web applications with complex AJAX-based scenarios.
WebDriver	• To use a certain programming language in designing your test case. • To test applications that are rich in AJAX-based functionalities. • To execute tests on the HtmlUnit browser. • To create customized test results.
Selenium Grid	• To run your Selenium RC scripts in multiple browsers and operating systems simultaneously. • To run a huge test suite, that needs to complete in the soonest time possible.

2.8.6 Advantages of Selenium

i) It is an Open source Software.

ii) It supports various Operating environments (Windows, Linux, Mac etc…)

iii) It supports various browsers (IE, Mozilla Firefox, Chrome, safari, Opera etc…)

iv) It supports various programming environments (Java, Perl, Python, Ruby and PHP)

v) It supports parallel Test execution.

vi) It uses less Hardware resources.

2.8.7 Disadvantages of Selenium

i) It supports Web based Applications only.

ii) No reliable support from anybody.

iii) No centralized maintenance of Elements/objects

iv) Difficult to setup environment.

v) Difficult to use.

vi) Limited support for Image based testing.

vii) New features may not work properly.

viii) No other tool integration for test management & No built in Reporting facility.

2.8.8 SeleniumVersus UFT

Selenium	UFT / QTP
1) Open Source	Vendor tool, License is required.
2) Supports various OS Environments.	MS Windows only.
3) Supports various Programming Environments	VBScript only.
4) No Object Repositories	Local and Shared object Repositories.
5) No built-in Reporting feature.	Built-in reporting feature.
6) Selenium WebDriver has no IDE and Selenium IDE has no Programming Interface.	UFT has both IDE and Programming Interface.
7) Uses less Hardware resources.	Uses more Hardware resources
8) Difficult to setup environment and use.	Easy to setup and use.
9) Limited support for Image Testing	Rich support for Image Testing
10) No Reliable support	Support from HP
11) No other tool integration for Test management.	UFT can be integrated with ALM/QC for Test Management.
12) New features may not work properly.	New features will properly.
13) No Add ins for supporting Application Environments.	Add ins are required for supporting Application environments.
14) Supports Web Applications only	Supports Desktop and Web Applications.
15) No Authorized Certification	Authorized Certification program.

2.8.9 What is TestNG?

TestNG is a powerful testing framework, an enhanced version of JUnit which was in use for a long time before TestNG came into existence. NG stands for 'Next Generation'.

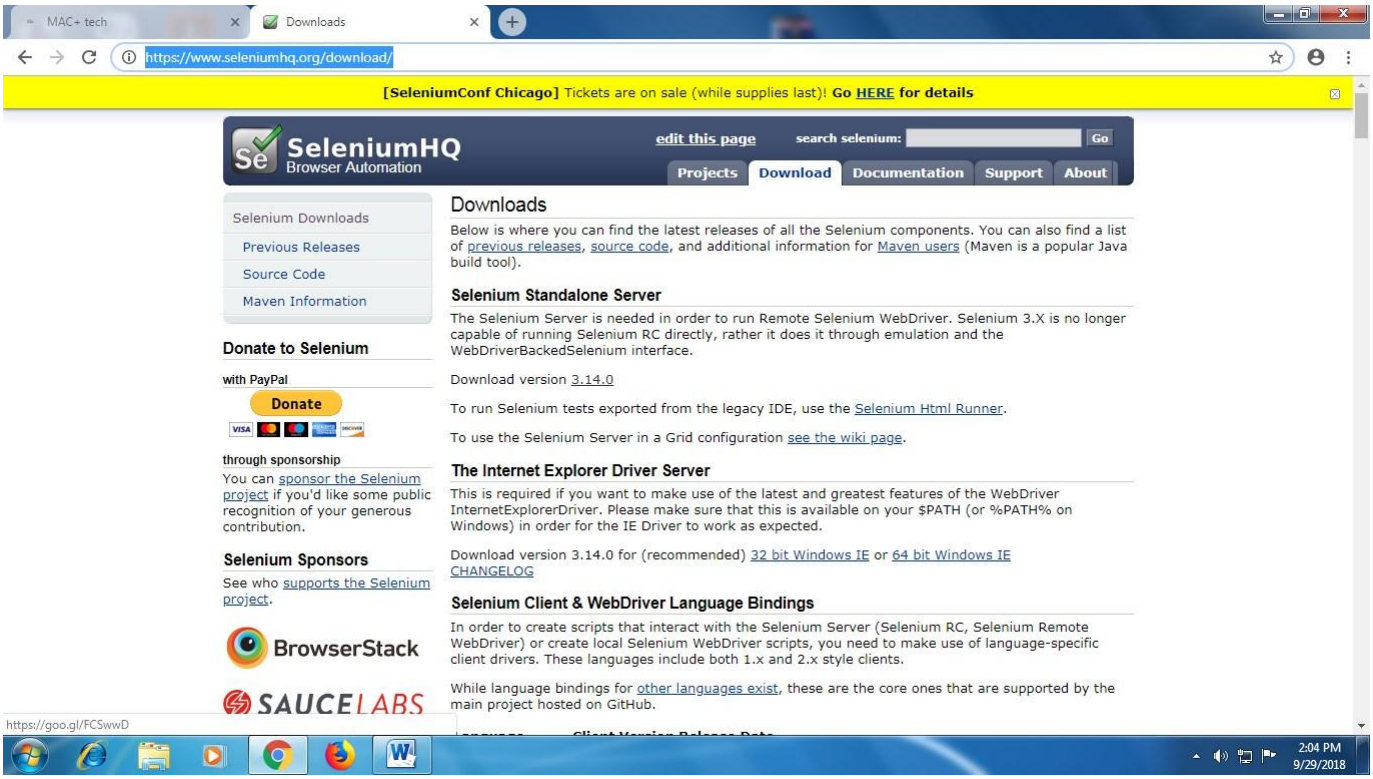
TestNG framework provides the following features −

- Annotations help us organize the tests easily.
- Flexible test configuration.
- Test cases can be grouped more easily.
- Parallelization of tests can be achieved using TestNG.
- Support for data-driven testing.
- Inbuilt reporting.

2.8.9 Step by Step Tutorial

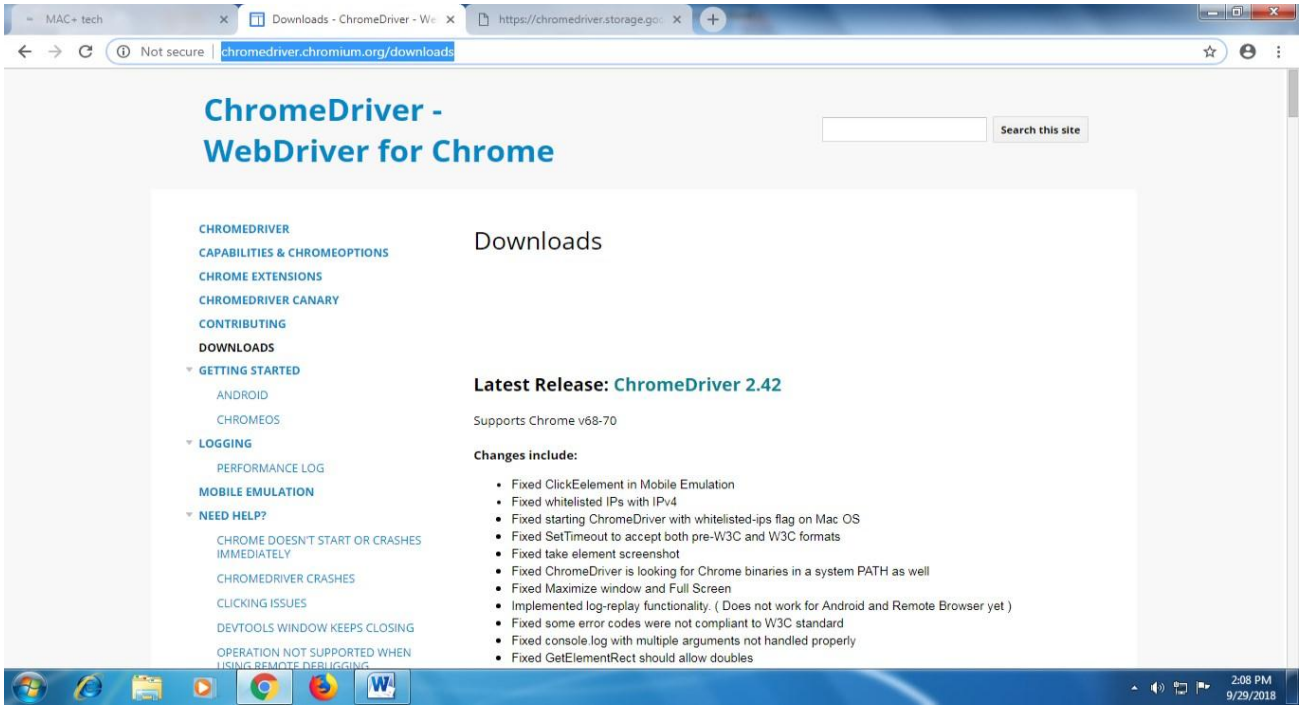
1. First of Download Latest Eclipse java photon-R version.
2. Download latest selenium-server-standalone-3.13.0 jar File from following link

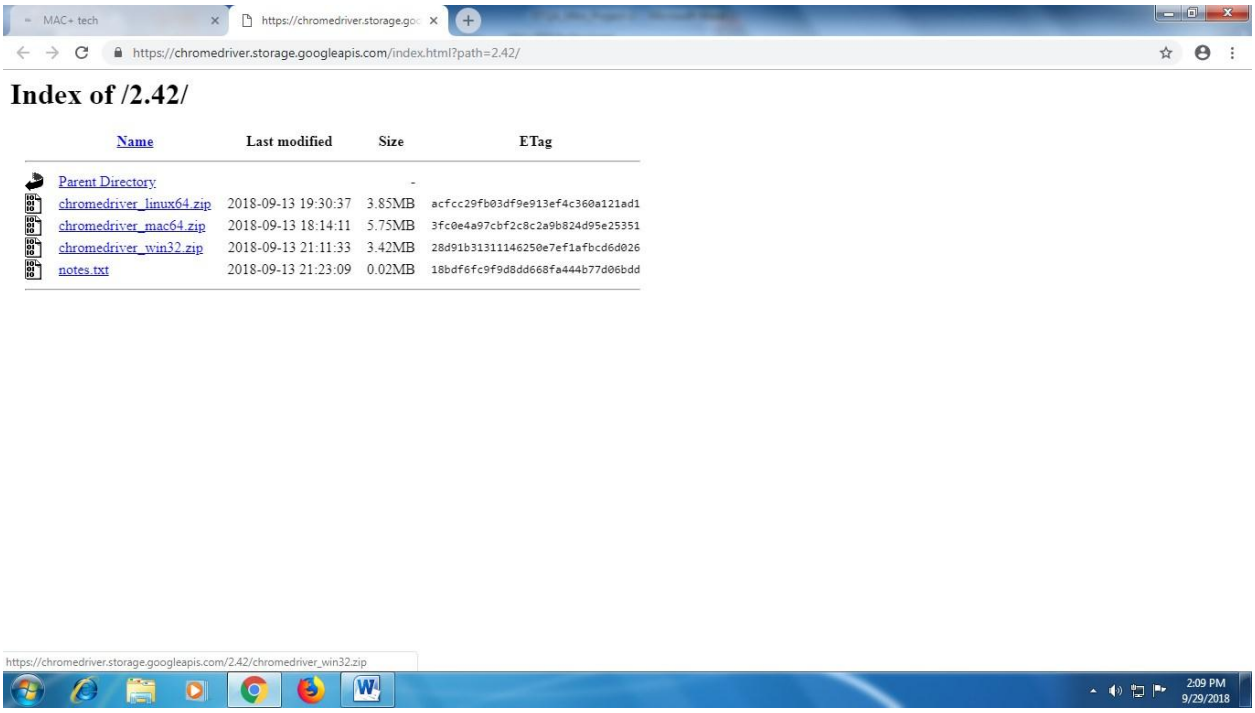
https://www.seleniumhq.org/download/ here on site 3.14.0 version is latest



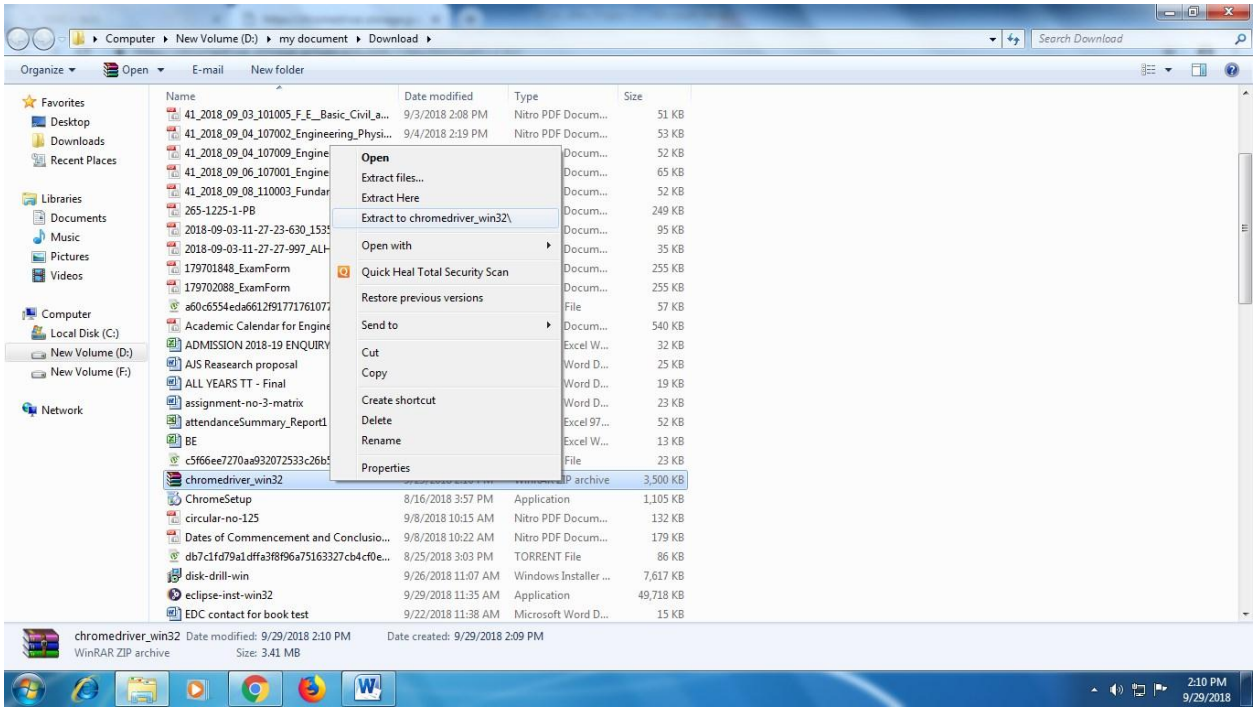
3. Download and Extract Chromedriver.exefor windows on any drive of computer.

http://chromedriver.chromium.org/downloads**here 2.42 is latest version**

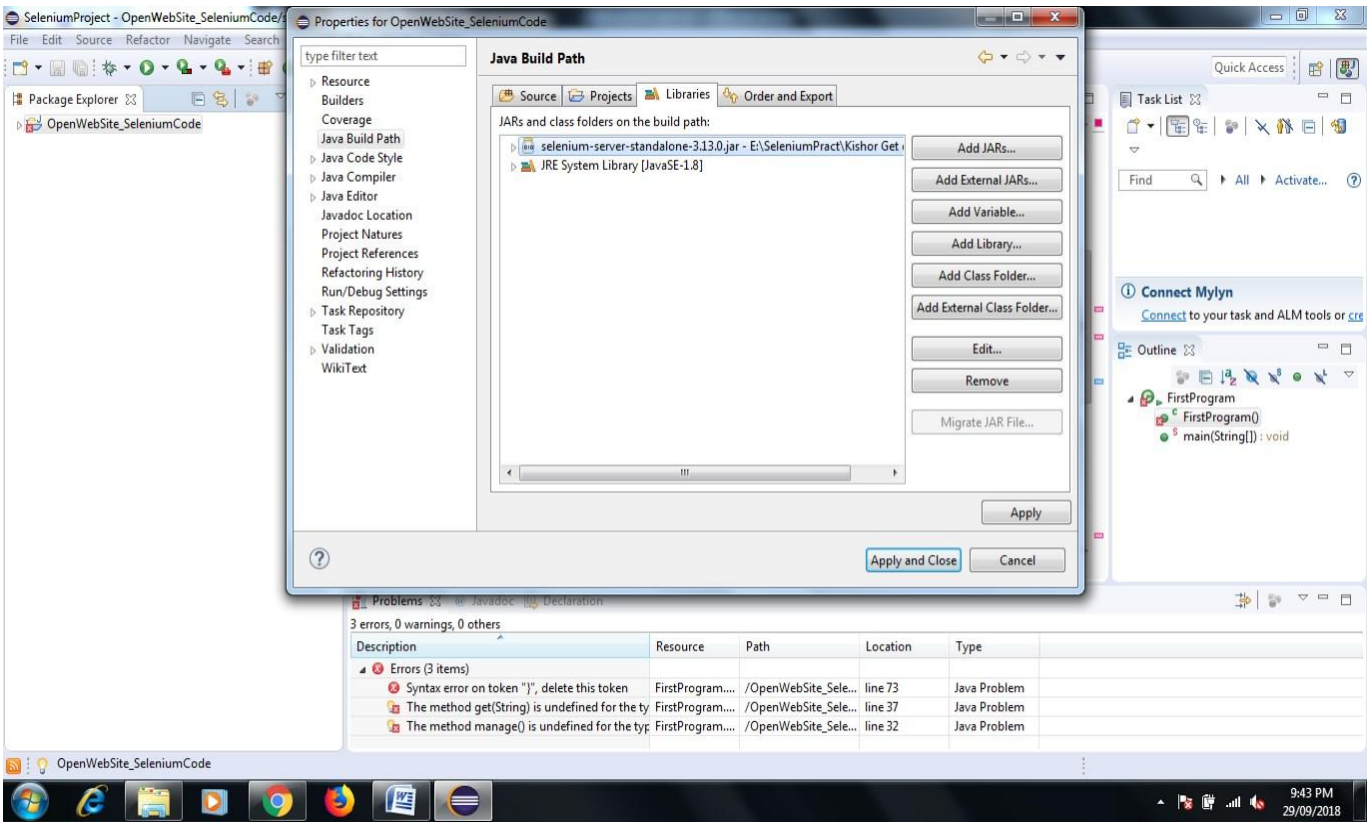




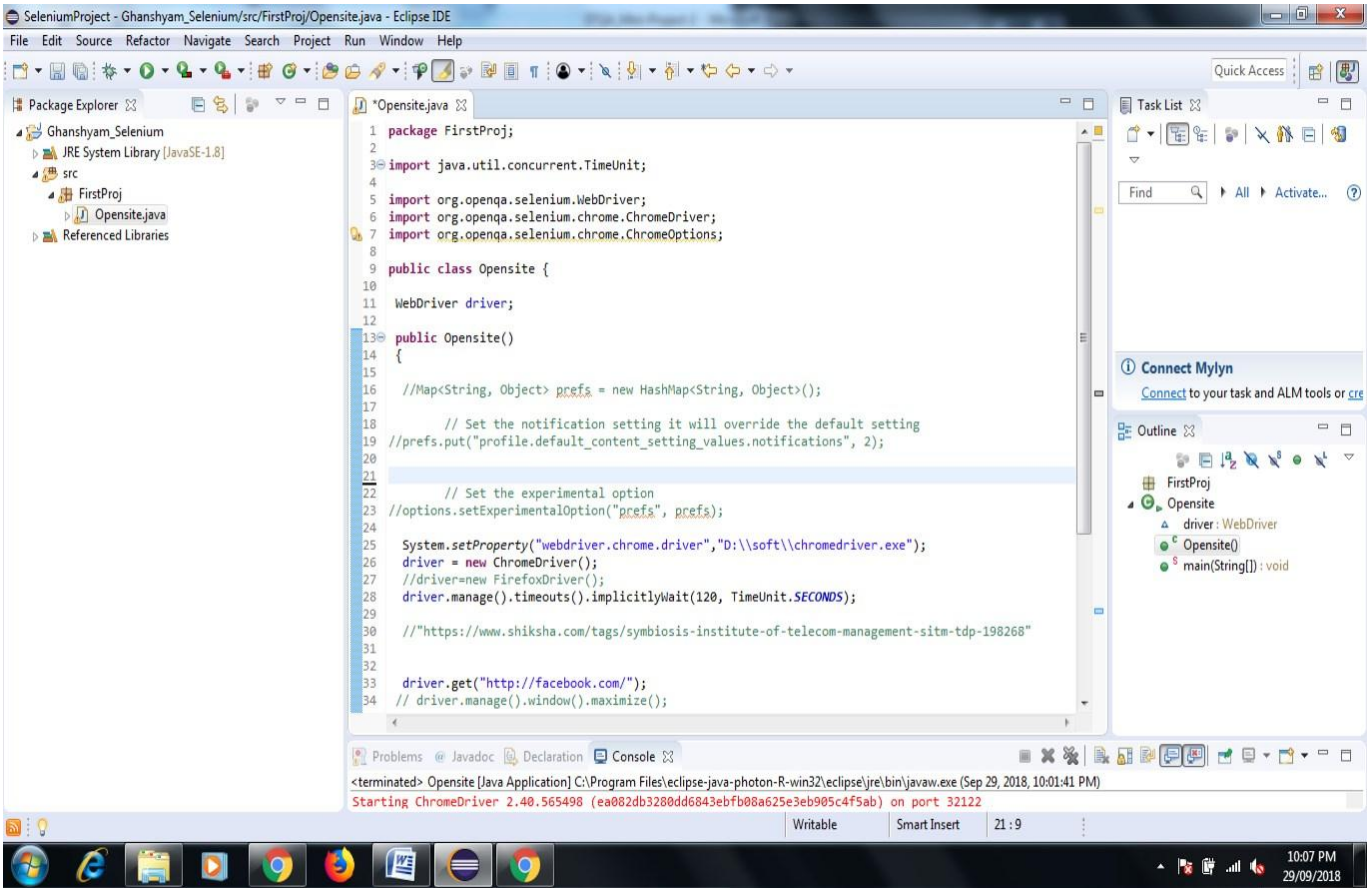
4. After Download Extract same on any Drive here I m extract on D Drive and my path of that exe file is **D:\my document\Download\chromedriver_win32**



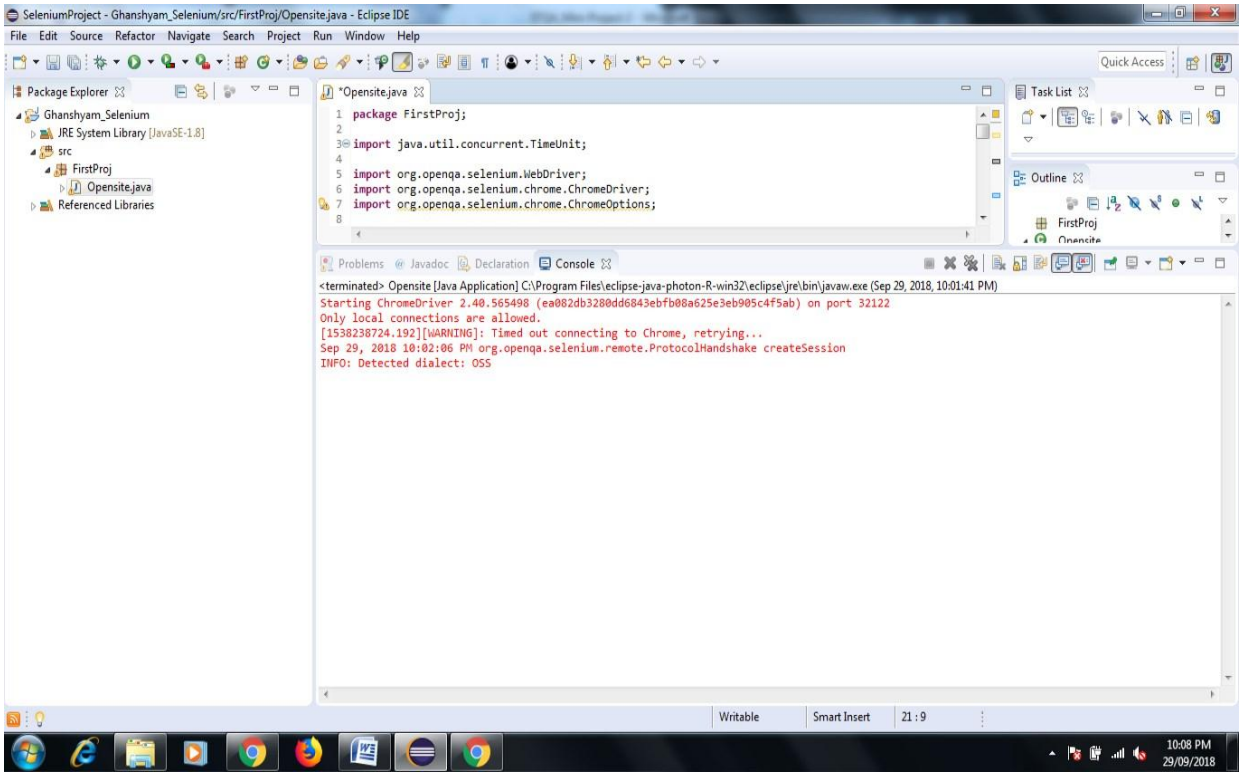
5. Now Open Eclipse IDE-----> Create Java Project-→ Right Click Project Name→Properties→ Java Build Path→Libraries→Add External JAR→ add selenium-server-standalone-3.13.0 jar →Apply and Close.



6. Now Want **Open the chrome browser with facebook.com page via Selenium Web driver Java Coding** so here we need to write Java Code in Class file which we already created
7. Here in My Program I Create **Ghanshyam_Selenium Java Project Folder name** and **Opensite.java is my class file** so write java code in this class file

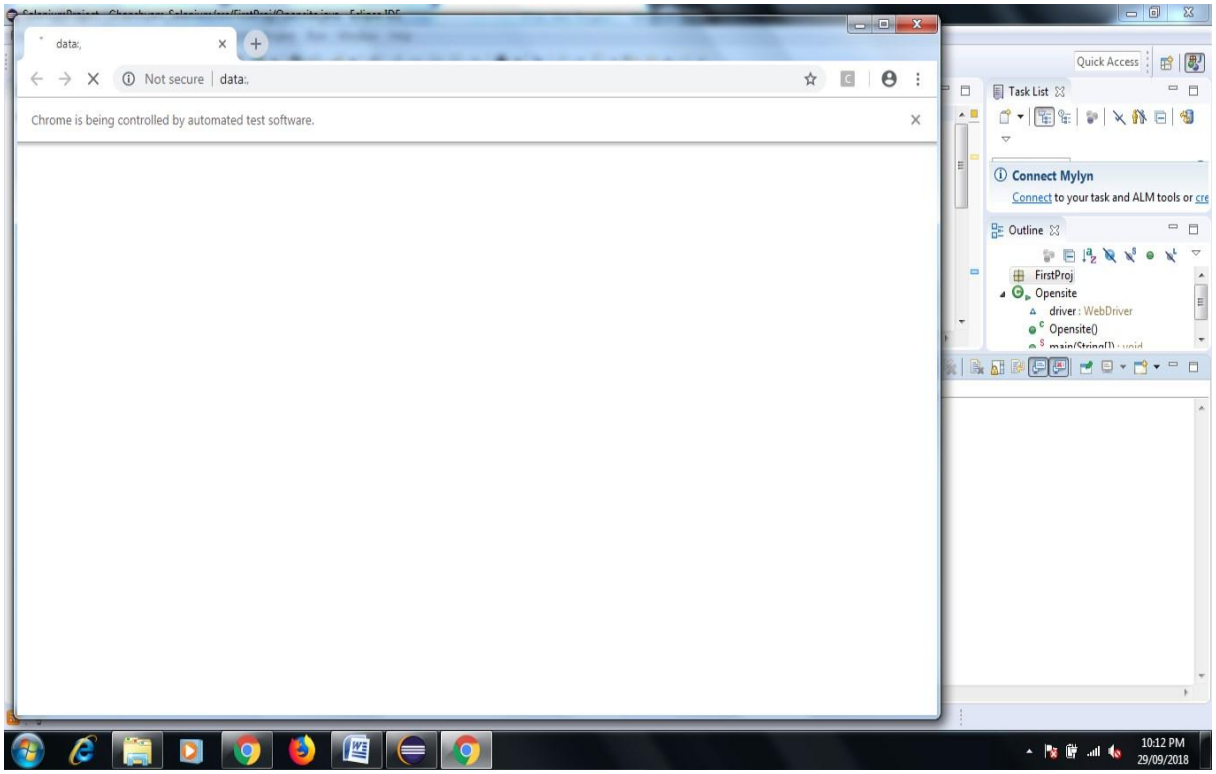


8. Right click on java program, select Run As and > "Java Application". After Code Successfully Run now see the output in Console Prompt



9. Now Your browser Open Automatically it shown data;
10. one more message display **chrome is being controlled by Automated Test Software**

It mean we open chrome browser and facebook.com page by selenium web driver java code Successfully.



In this way Our First Module Run Successfully.

Module-2 In This Project I want to collect all Rating Feedback related our college available on website Justdial.com

1. Create One Java Project Folder Give Name→Review Demo→Now Create One Class File give name Practo.com
2. Right Click Project Name→Properties→ Java Build Path→Libraries→Add External JAR→ add selenium-server-standalone-3.13.0 jar →Apply and Close.
3. Similarly Copy the Chromedriver.exe file on my D Drive `D:/soft/chromedriver.exe` in this way.
4. Now Start right writing code for Practo.java to Collect all review of any Hospital.

Here I want search all feedback of **SNJB-s-Late-Sau-Kantabai-Bhavarlalji-Jain-College-Of-Engineering-Neminagar-Chandwad**

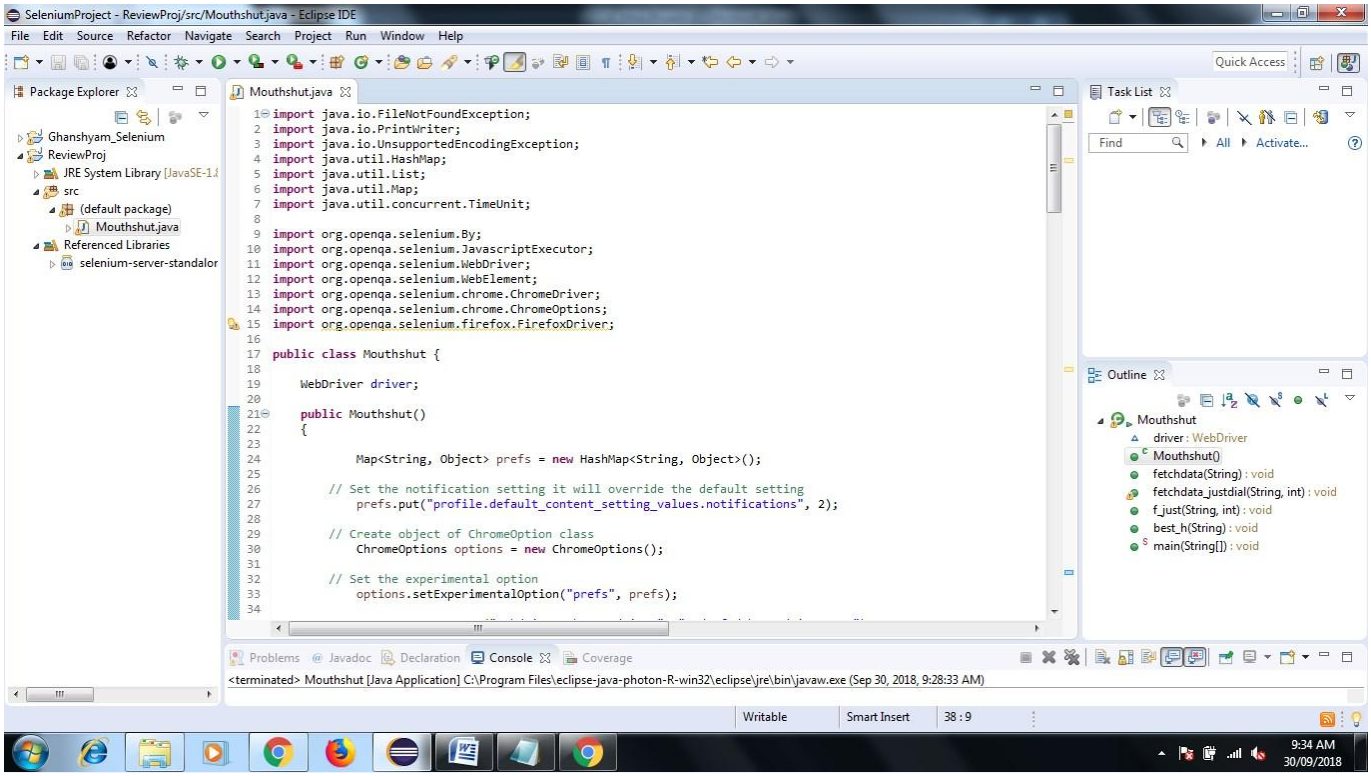
Link of Feedback of All Customer-

https://www.justdial.com/Nashik/SNJB-s-Late-Sau-Kantabai-Bhavarlalji-Jain-College-Of-Engineering-Neminagar-Chandwad/0253PX253-X253-151019113056-B2P9_BZDET/reviews/page-6

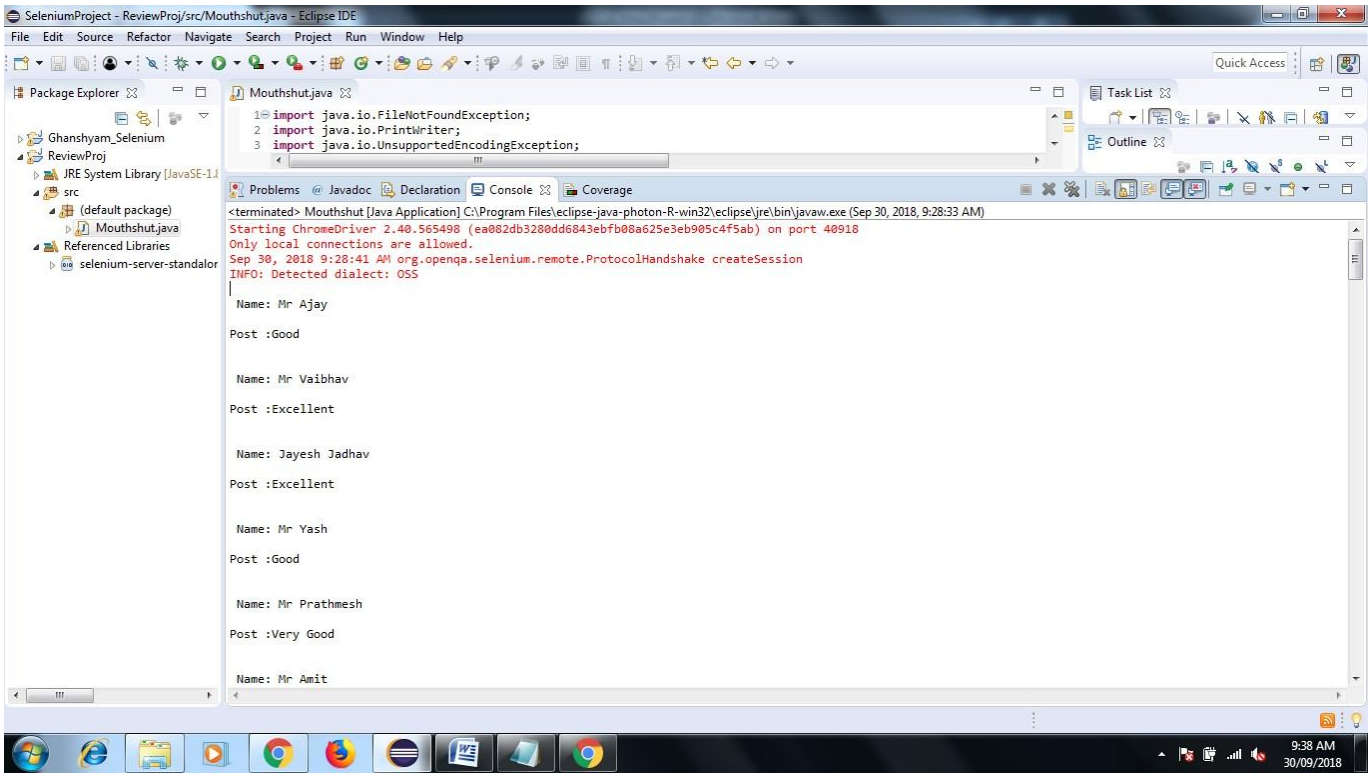
5. Now to save Feedback of all pages here I create one text file give name C:\Users\admin\eclipse-workspace\SeleniumProject\ReviewProj\Snjb.txt

Now First of All Execute Code here my file name Mouthshut.java

Mouthshut.com is website like Justdialcom

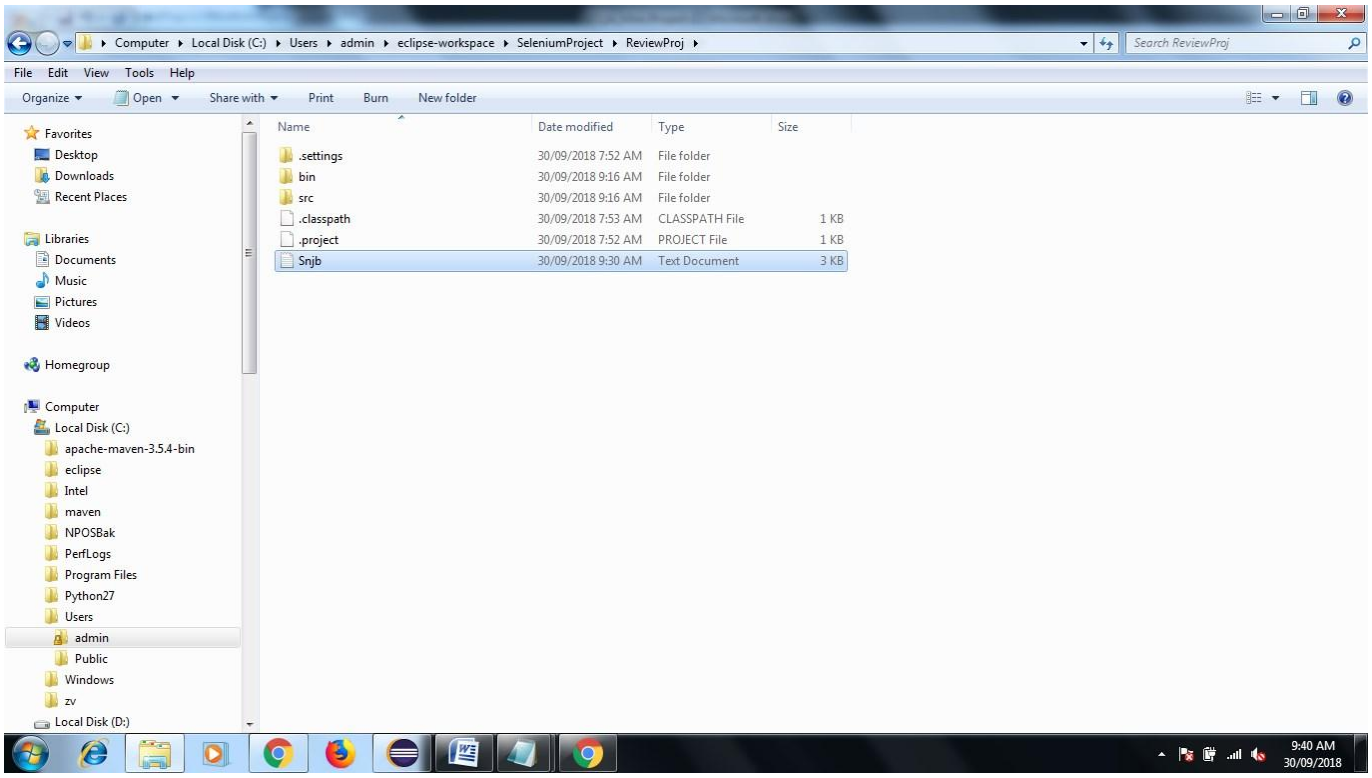


After Execution Code Mouthshut.java Chrome Browser Opened Automatically with Specified website you can also see the output of rating in console as well as file you created.

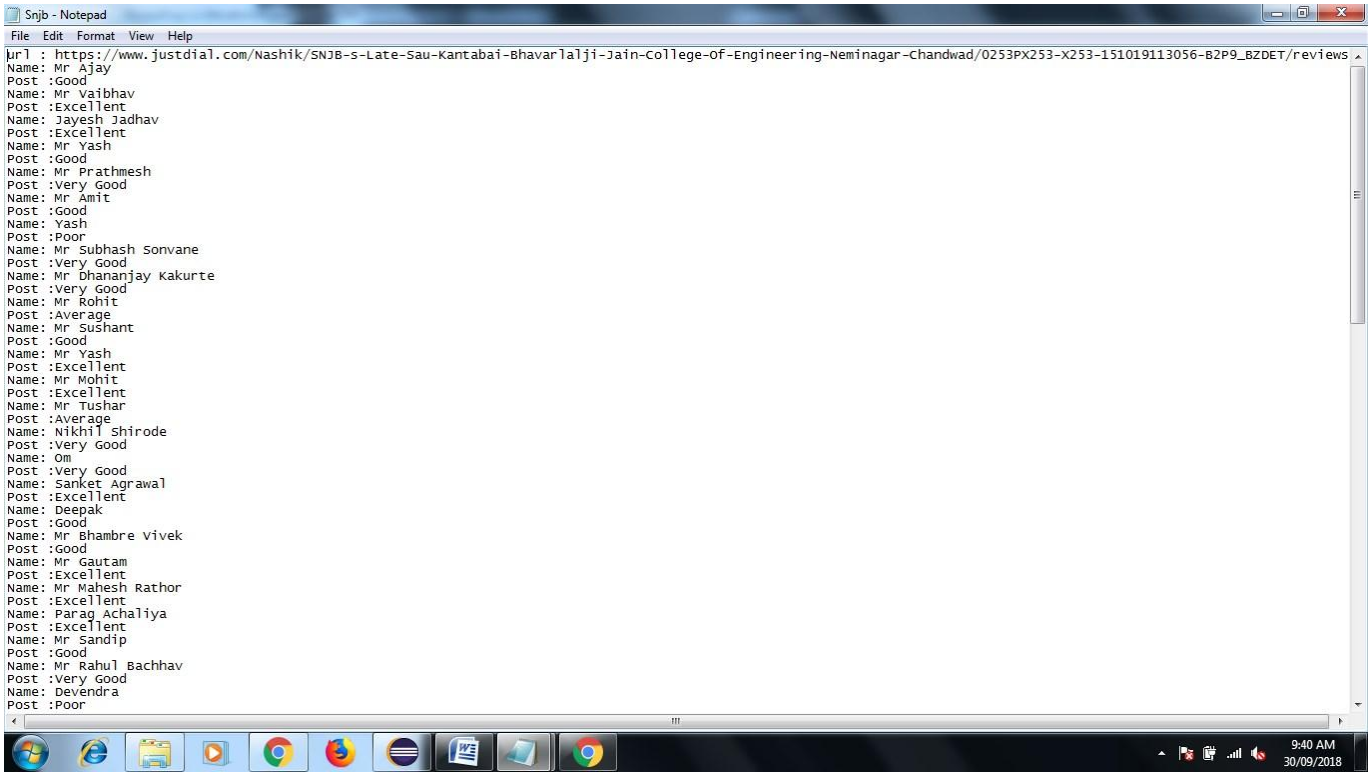


Now Lets Check the Feedback of All Customer go to your Eclipse Workspace Path my path is

C:\Users\admin\eclipse-workspace\SeleniumProject\ReviewProj\Snjb.txt



After Open Txt File See the Output



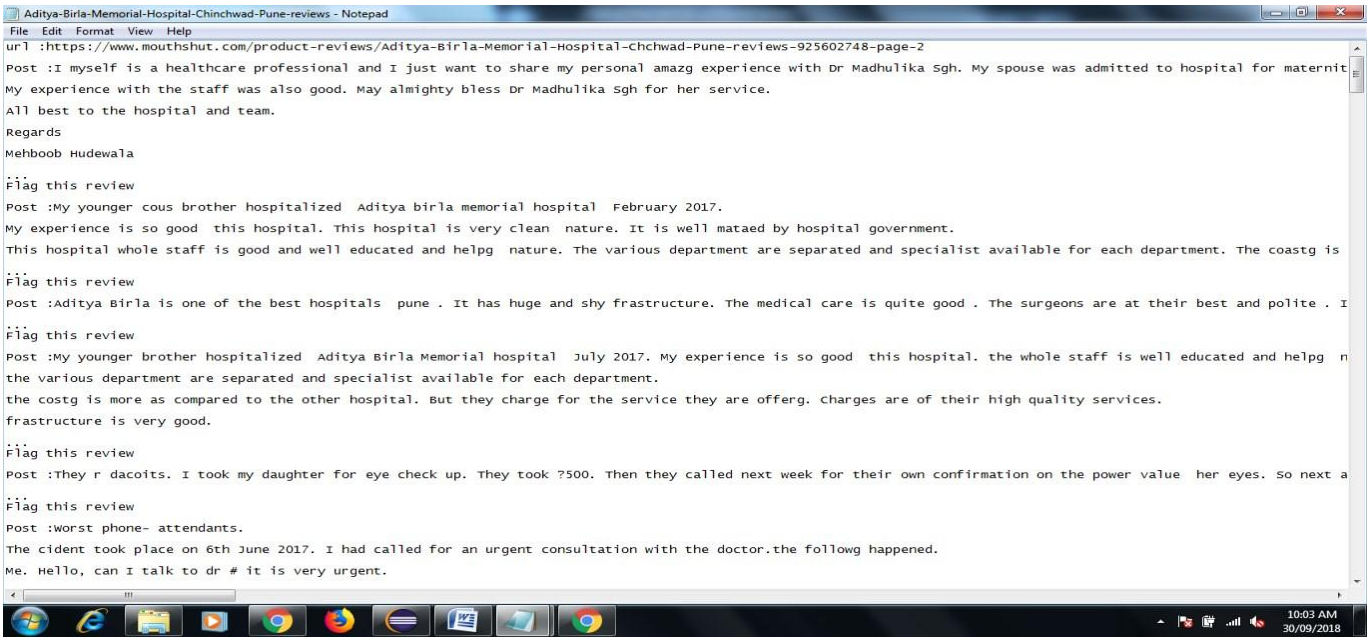
In This you can Collect all different kind of colleges, hospital rating and review from various website like mouthshout.com,Justdial.com via Selenium Web driver Tool

---

Module-3 Now I want Collect Review of Aditya-Birla-Memorial-Hospital-Chinchwad-Pune

Link- https://www.mouthshut.com/product-reviews/Aditya-Birla-Memorial-Hospital-Chchwad-Pune-reviews-925602748-page-2

After Execution of Code-

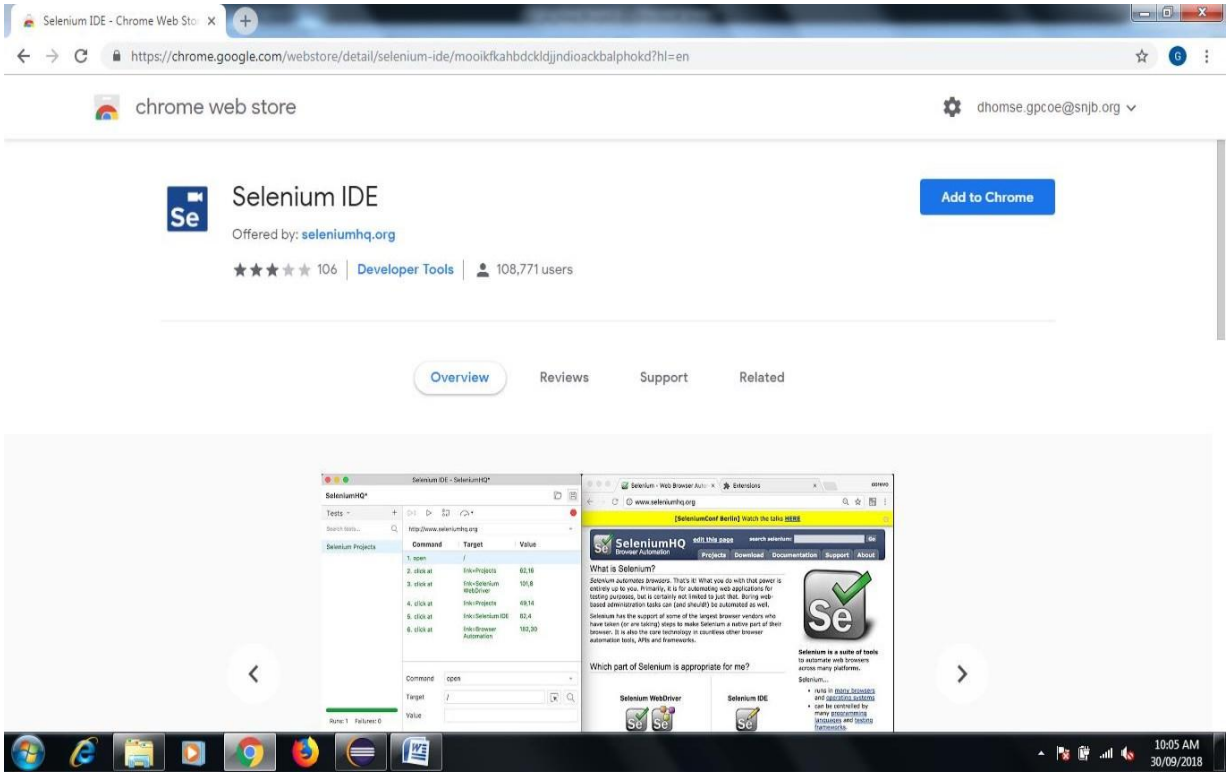


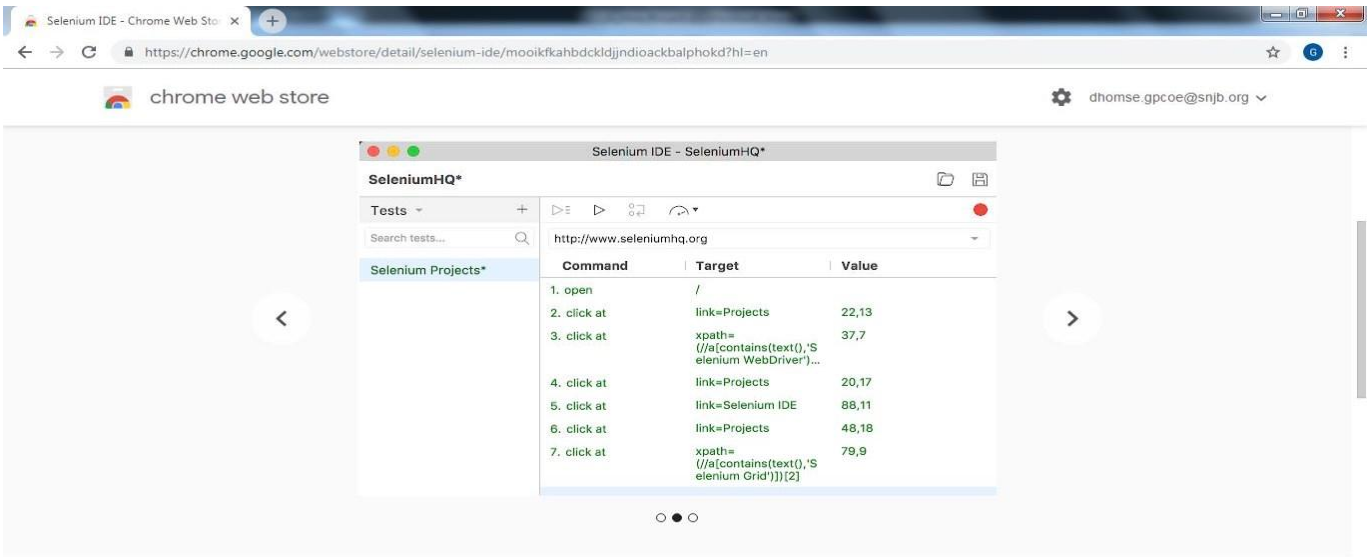
Selenium IDE:

1. **Download Selenium IDE Chrome Extension from following Link**

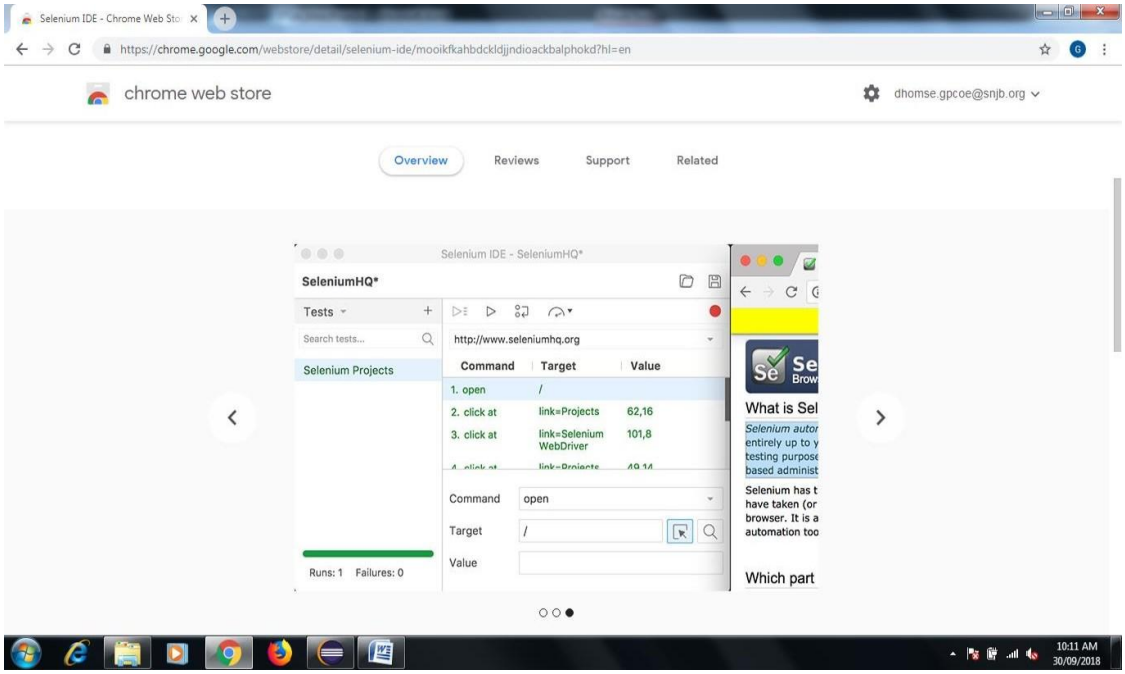
https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=en

2. **Click on Add to Chrome**
3. **Check the icon on of IDE Square in right side after browser after successfully installation to chrome**
4. **Now Click on record button do some operation on website…see the report in console of IDE command target and value….in this way you also find out xpath etc information.**





You can also check the Automated Test Case Pass or Fails see the following figure



2.9 Oral Question

1. What is the difference between Selenium and QTP?
2. What is mean by Bug taxonomy?
3. How to verify error and message in selenium webdriver?
4. What are the different types of drivers available in WebDriver?
5. What are the different types of locators in selenium?

2.10 Conclusion

In this way you learn how to use Selenium Open Source Tool for perform Automation Testing on web based application.