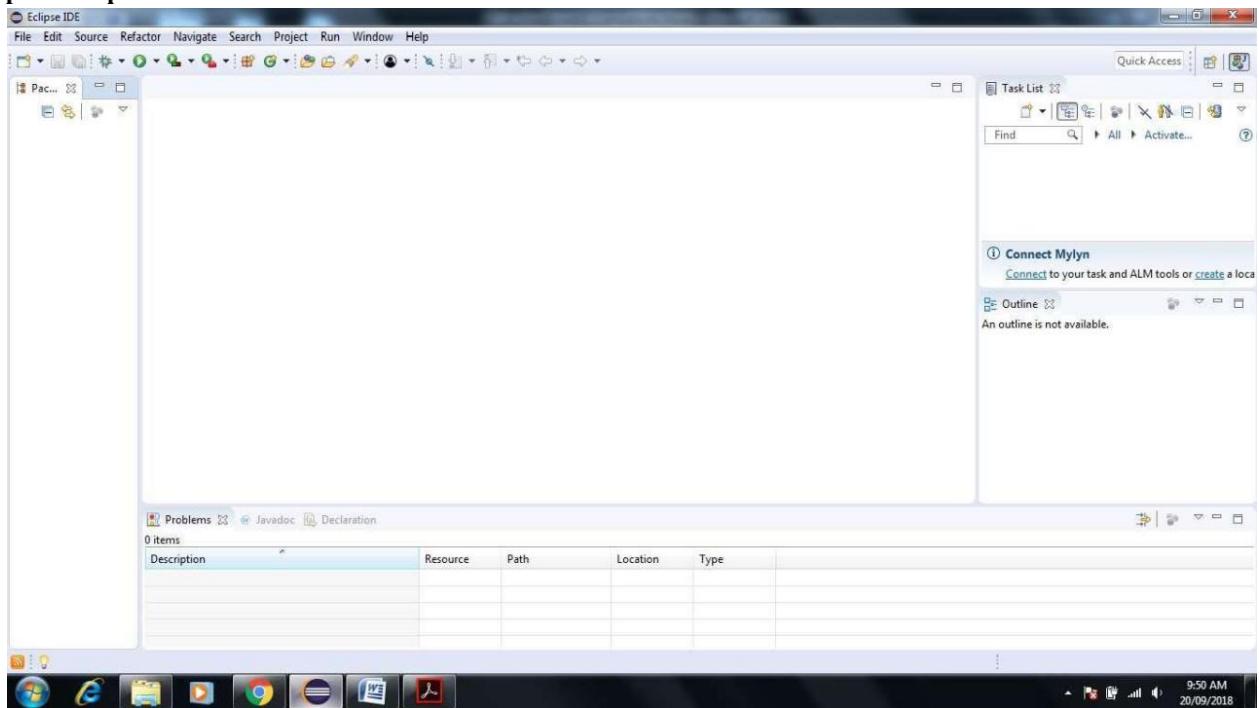
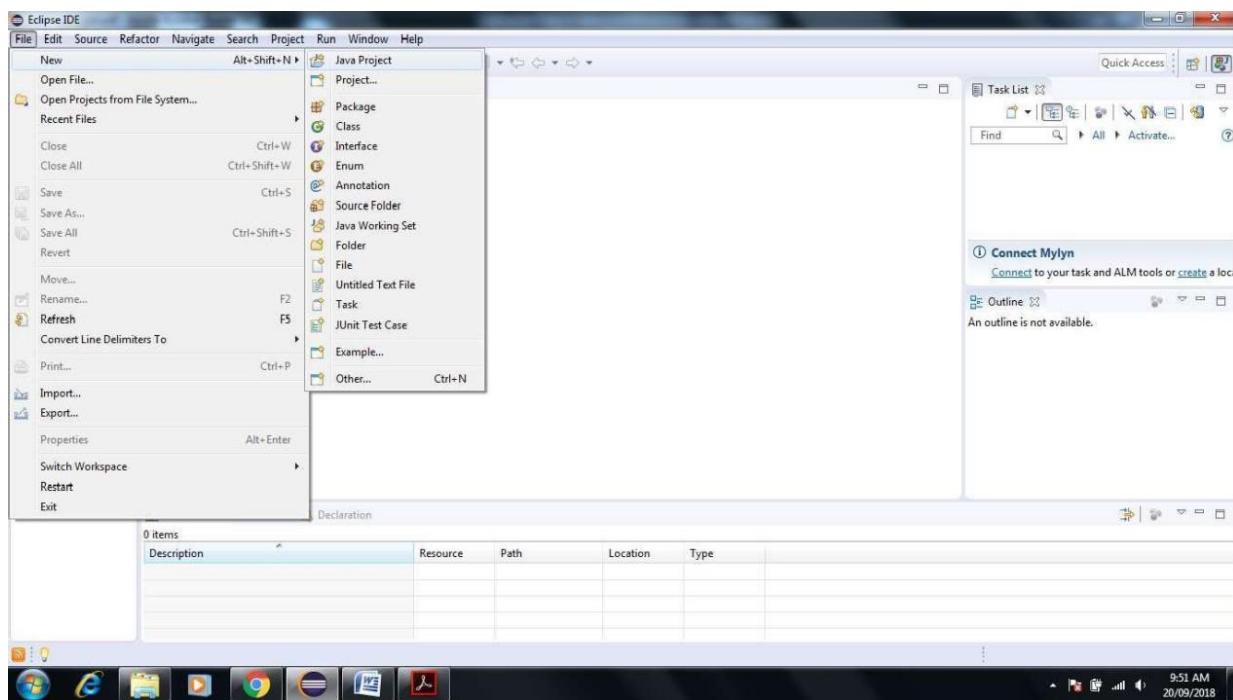


CHAPTER 3: IMPLEMENTATION

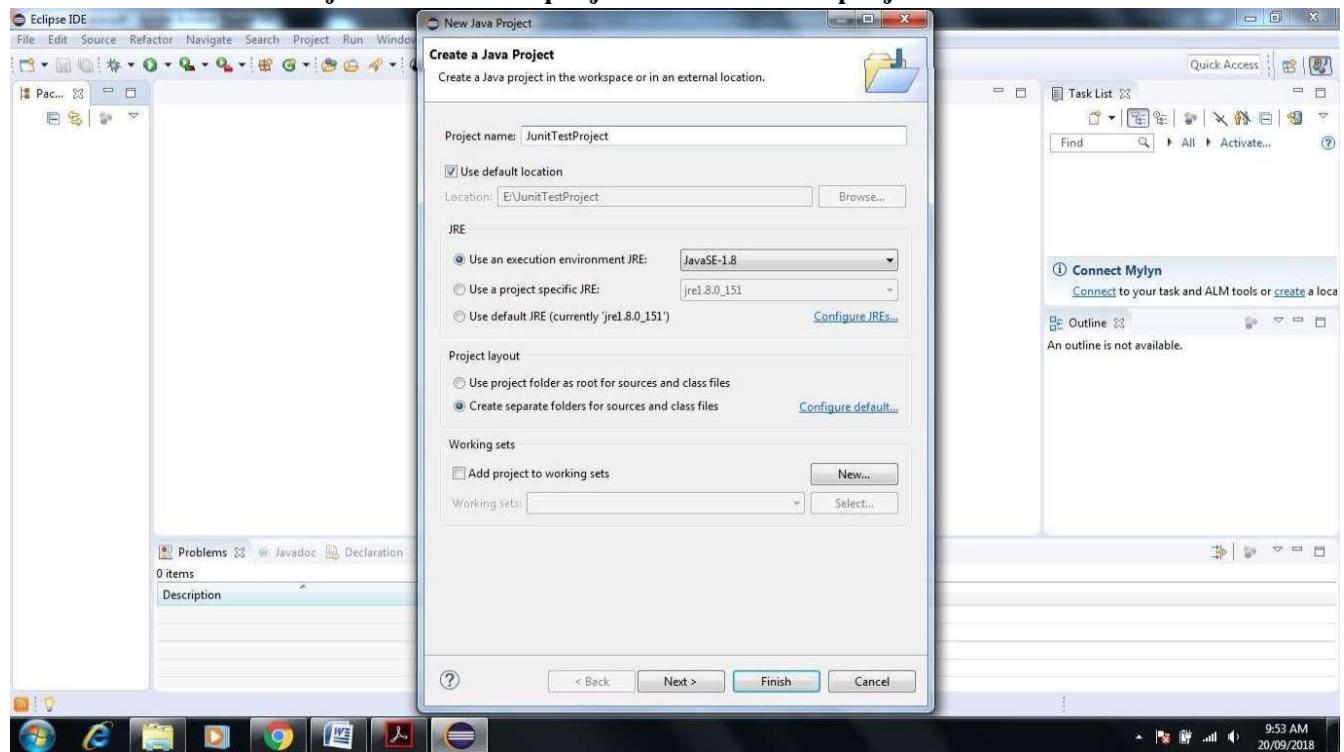
1. Download JDK 1.8 and Eclipse latest version here we are using eclipse-java-photon-R-win32.
2. Open Eclipse IDE



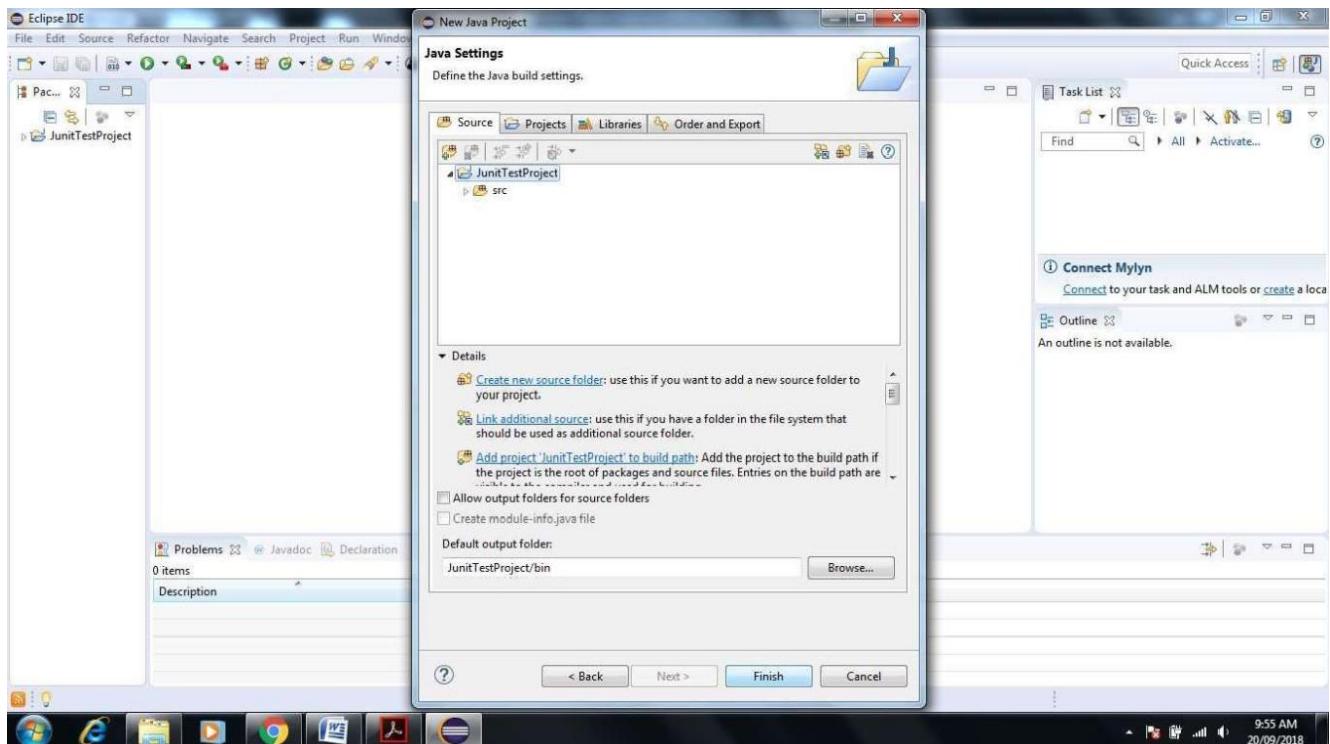
3. Go to File and Select New -> Create New Java Project



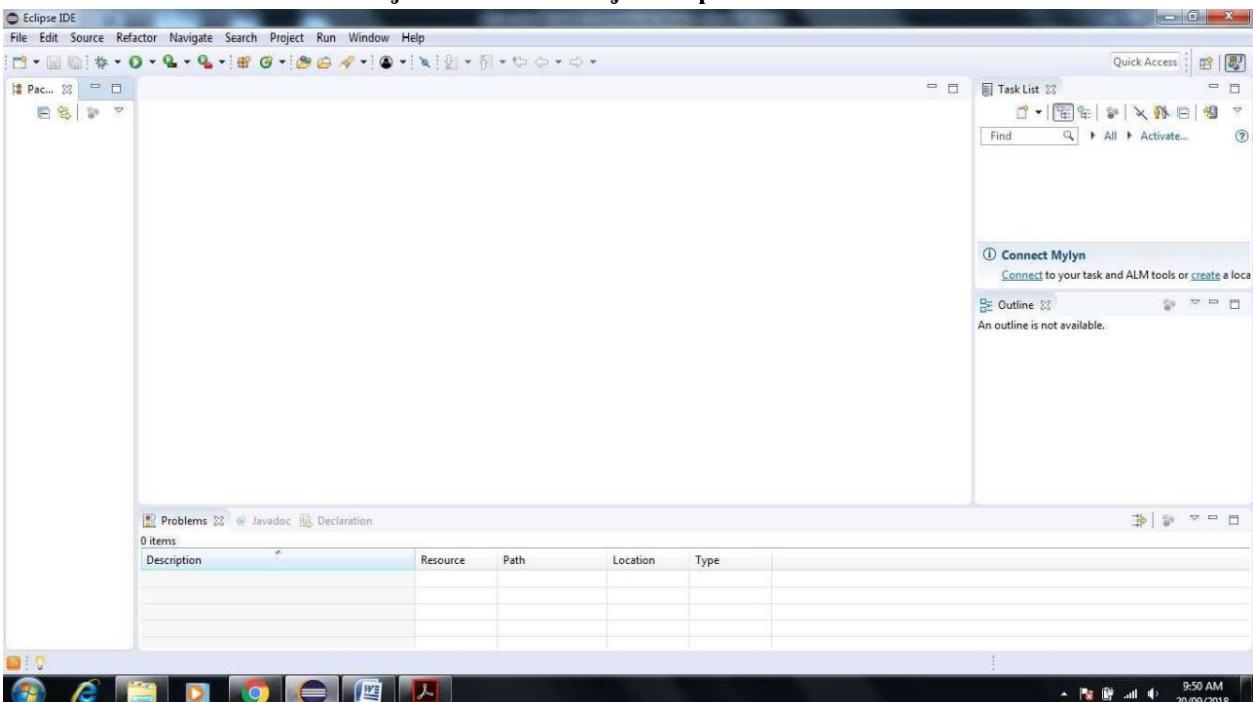
4. Give Junit Test Project name to the project and check use project folder as root for source and class files



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

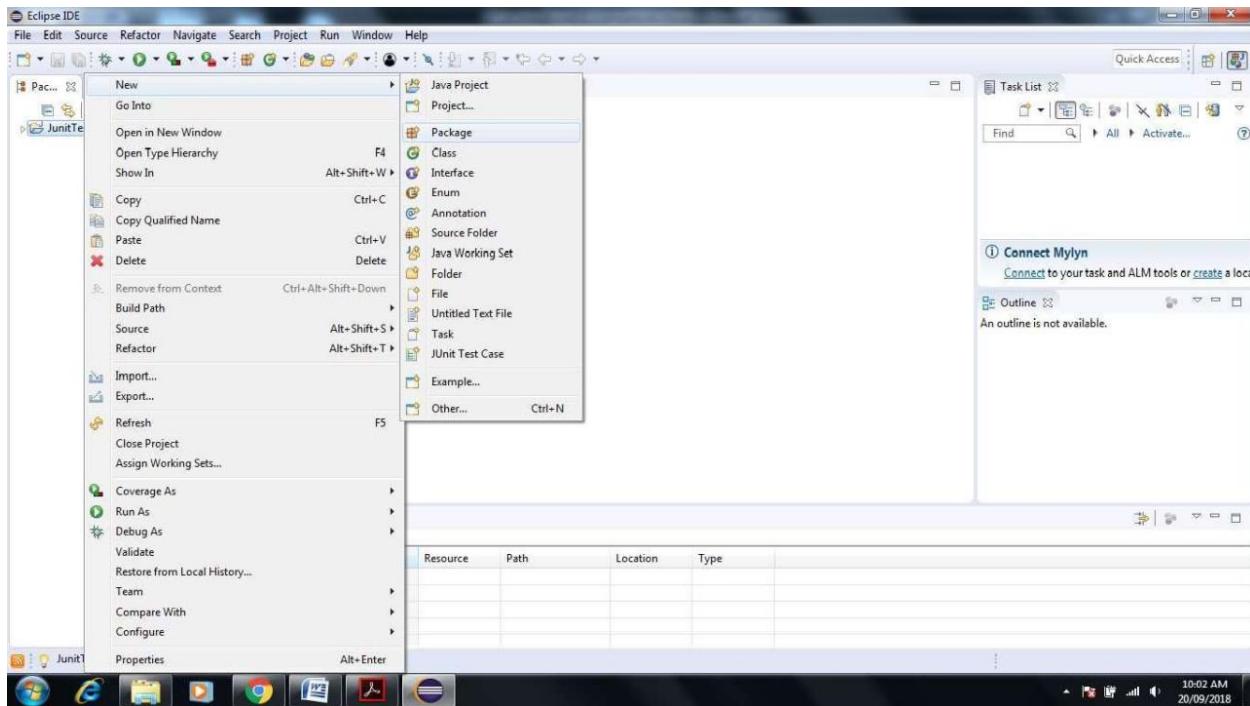


Next Screen Shown JunitTestProject Folder in Project Explorer

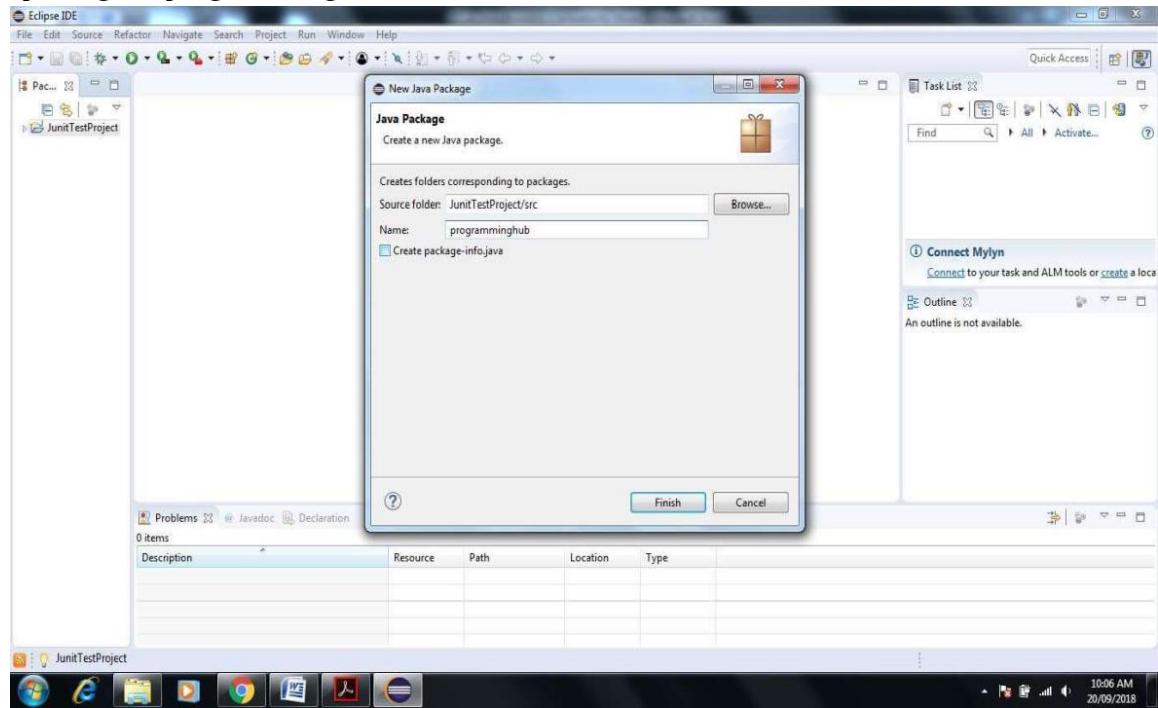


6.

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

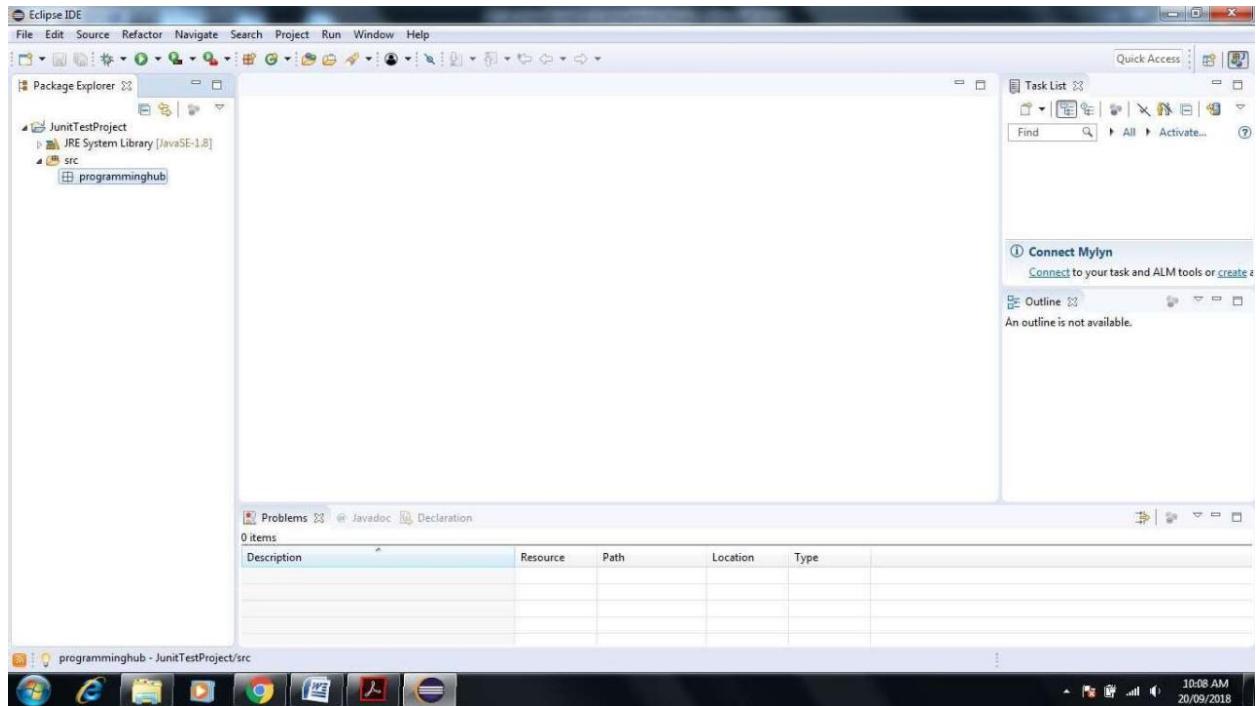


Name package as programming hub-> Click on Finish

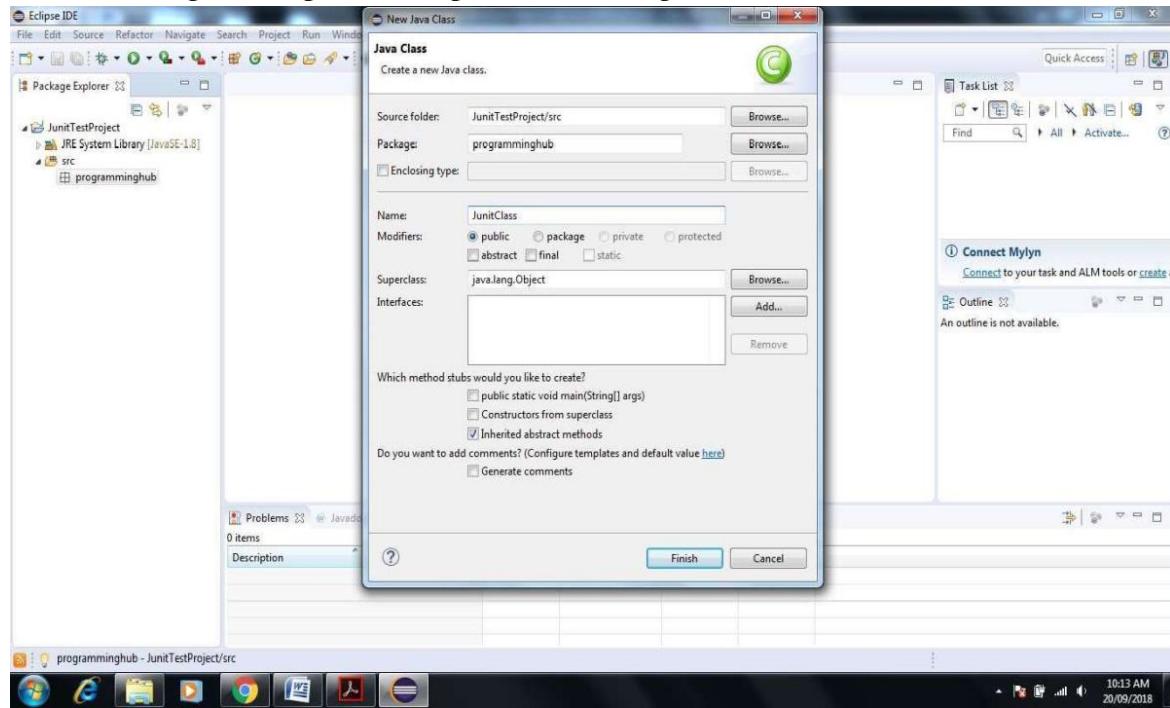


8.

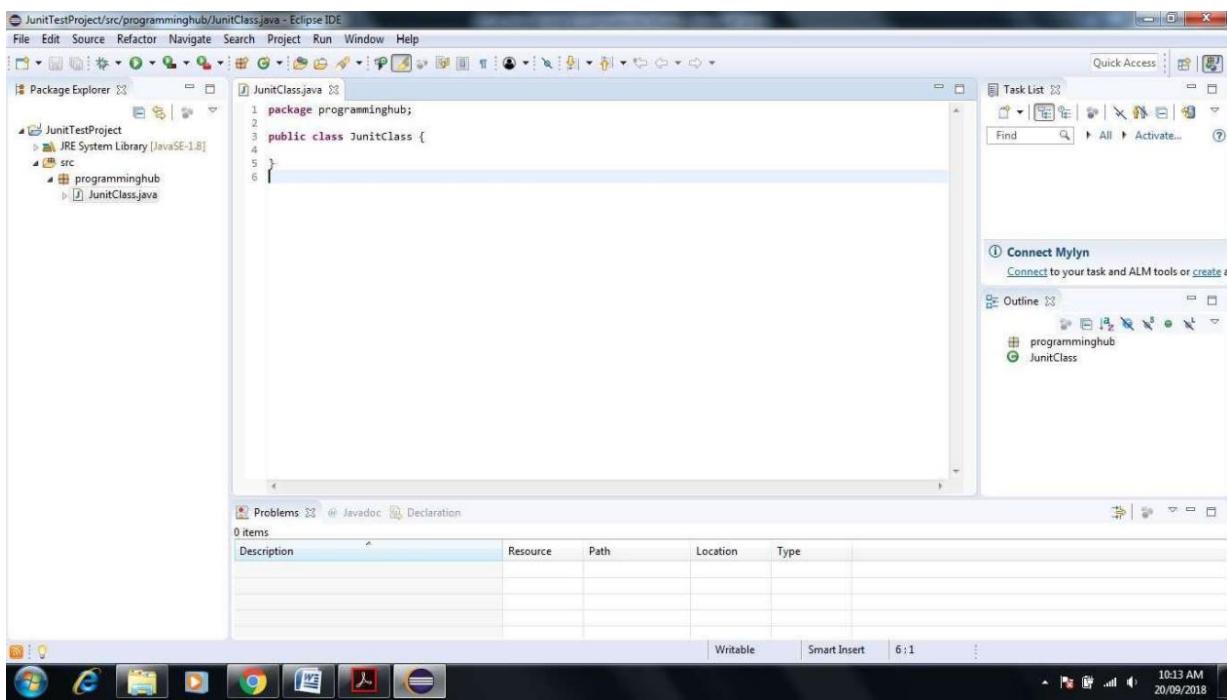
9. See the Programming hub package see in project Explorer Screen of Eclipse



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

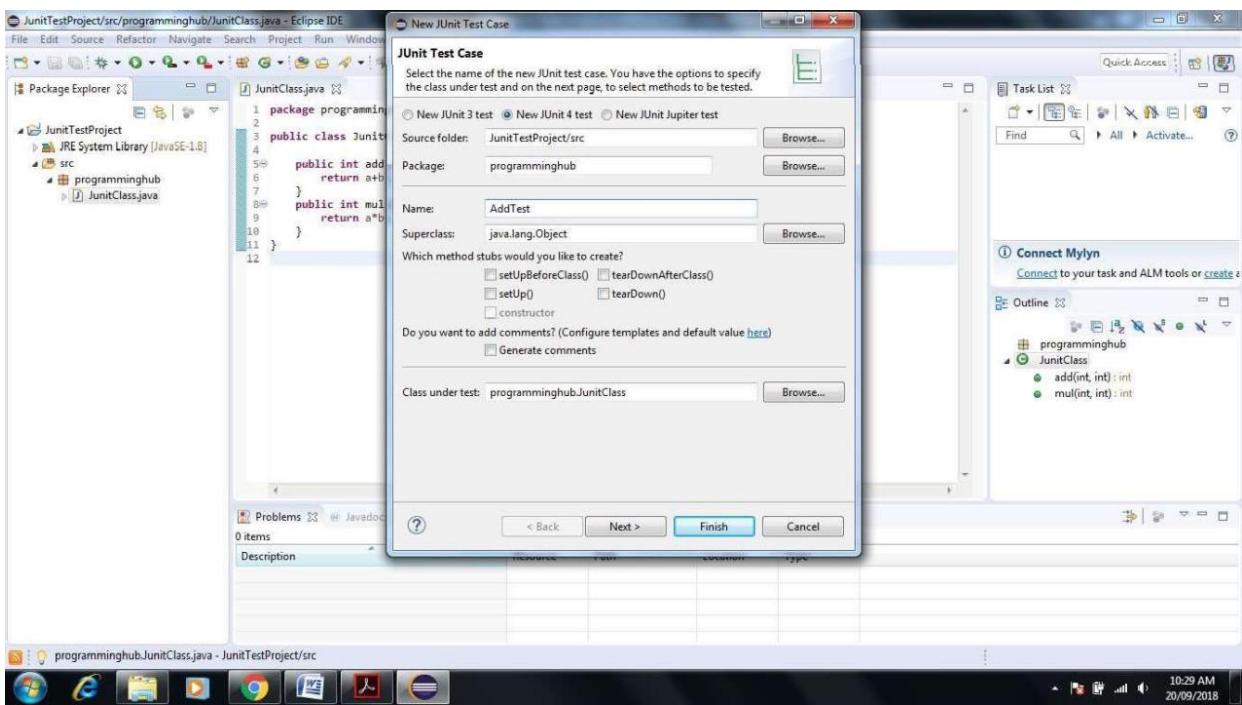
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project "JunitTestProject" with a package "src" containing "JunitClass.java".
- Code Editor:** Displays the code for `JunitClass.java`:

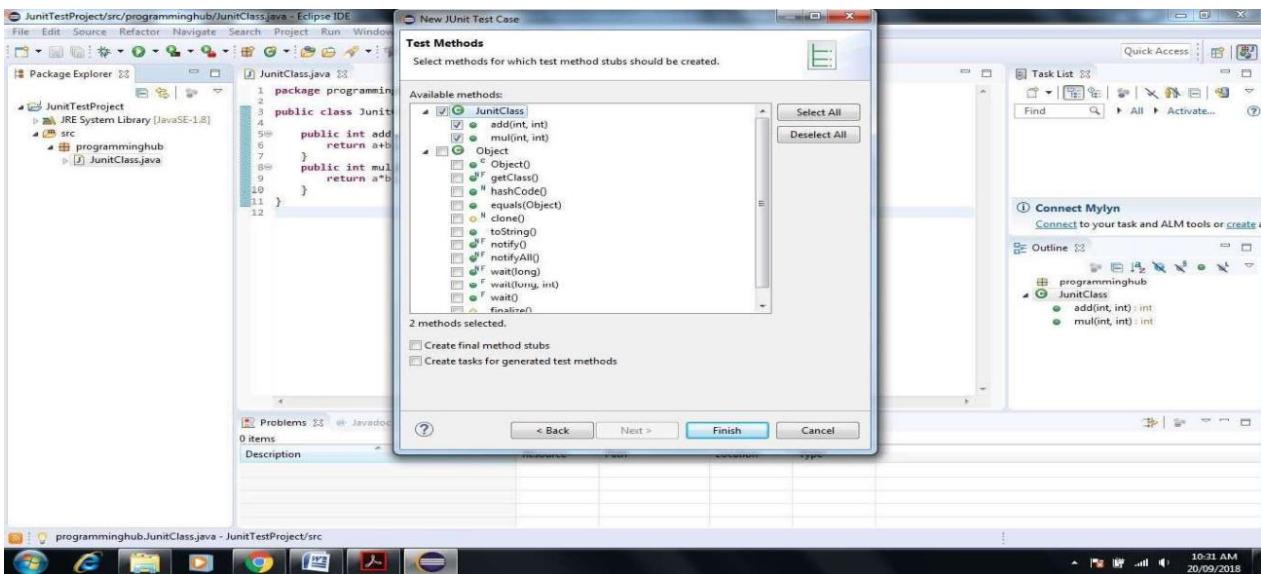
```
1 package programminghub;
2
3 public class JunitClass {
4     public int add(int a,int b) {
5         return a+b;
6     }
7     public int mul(int a,int b) {
8         return a*b;
9     }
10}
```
- Outline View:** Shows the class `JunitClass` with methods `add(int, int)` and `mul(int, int)`.
- Task List:** Empty.
- Problems:** Empty.
- Bottom Bar:** Shows the Windows taskbar with various icons and the system clock indicating 10:19 AM on 20/09/2018.

13. Write Test Cases for Java Program

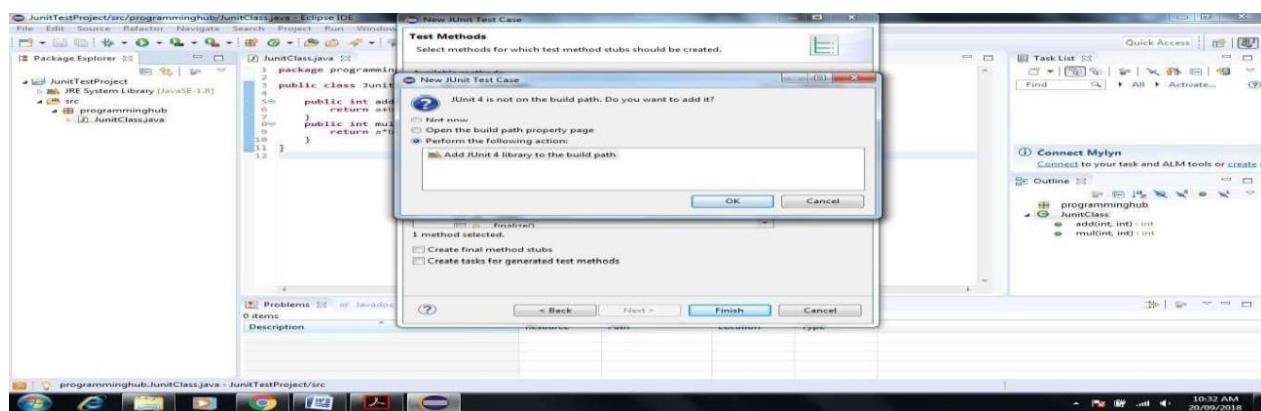
14. Name test suite as Add Test and choose New Junit4 test



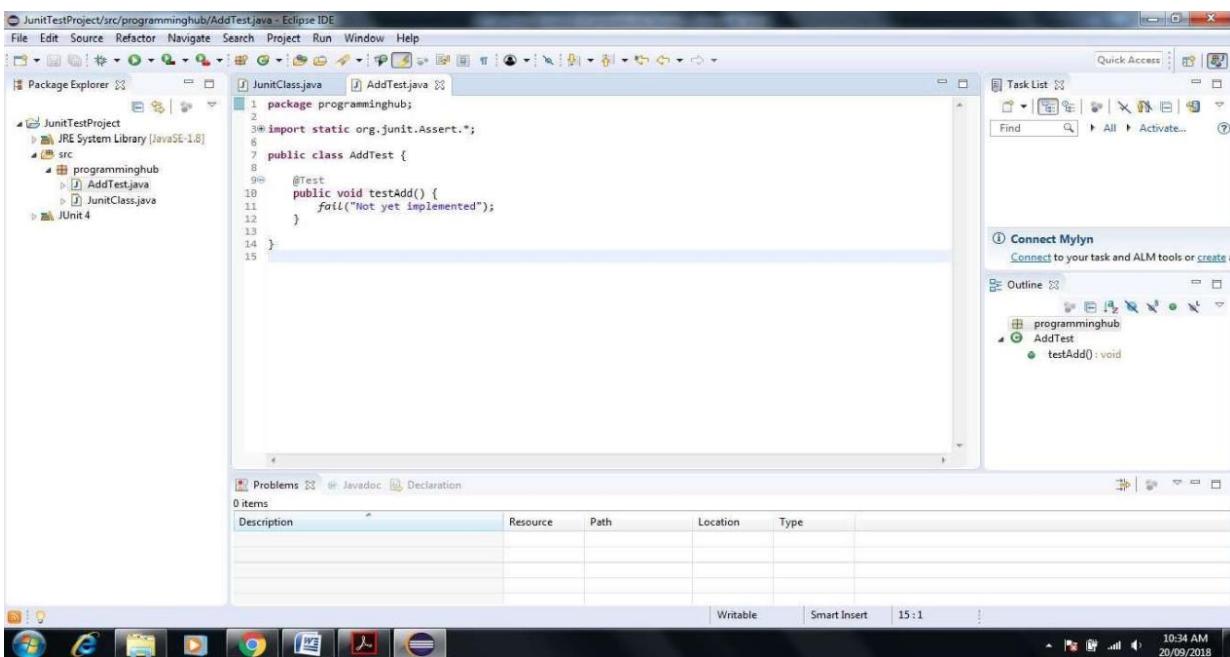
15. Click on add Checkbox



16. Click on Next-> Ok



17. Next screen will appear



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

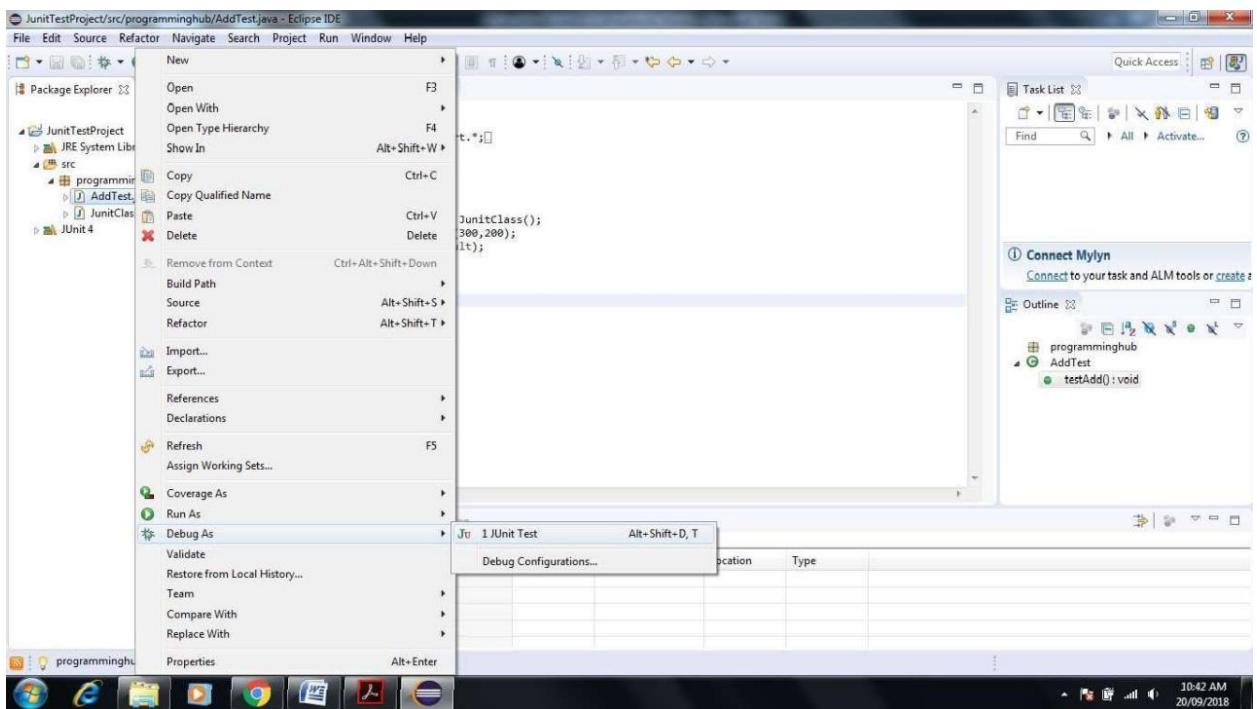
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a package named "programminghub" containing classes "AddTest.java" and "JunitClass.java".
- Editor:** Displays the content of "AddTest.java":

```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class AddTest {
6
7     @Test
8     public void testAdd() {
9         JunitClass junit=new JunitClass();
10        int result=junit.add(300,200);
11        assertEquals(500,result);
12    }
13
14 }
15
16 }
```

- Outline View:** Shows the class "AddTest" and its method "testAdd()".
- Problems View:** Shows 0 items.
- Task List:** Shows a message to "Connect Mylyn".
- Bottom Status Bar:** Shows the date and time: 10:39 AM 20/09/2018.

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)

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** JunitTestProject/src/programminghub/AddTest.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbars:** Standard toolbar with icons for file operations.
- Left Sidebar:** Package Explorer (selected), JUnit (disabled).
- Middle Area:** Editor showing AddTest.java code:

```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class AddTest {
6
7     @Test
8     public void testAdd() {
9         JUnitClass junit=new JUnitClass();
10        int result=junit.add(300,200);
11        assertEquals(500,result);
12    }
13
14 }
15
16 }
```
- Bottom Status Bar:** 10:44 AM, 20/09/2018

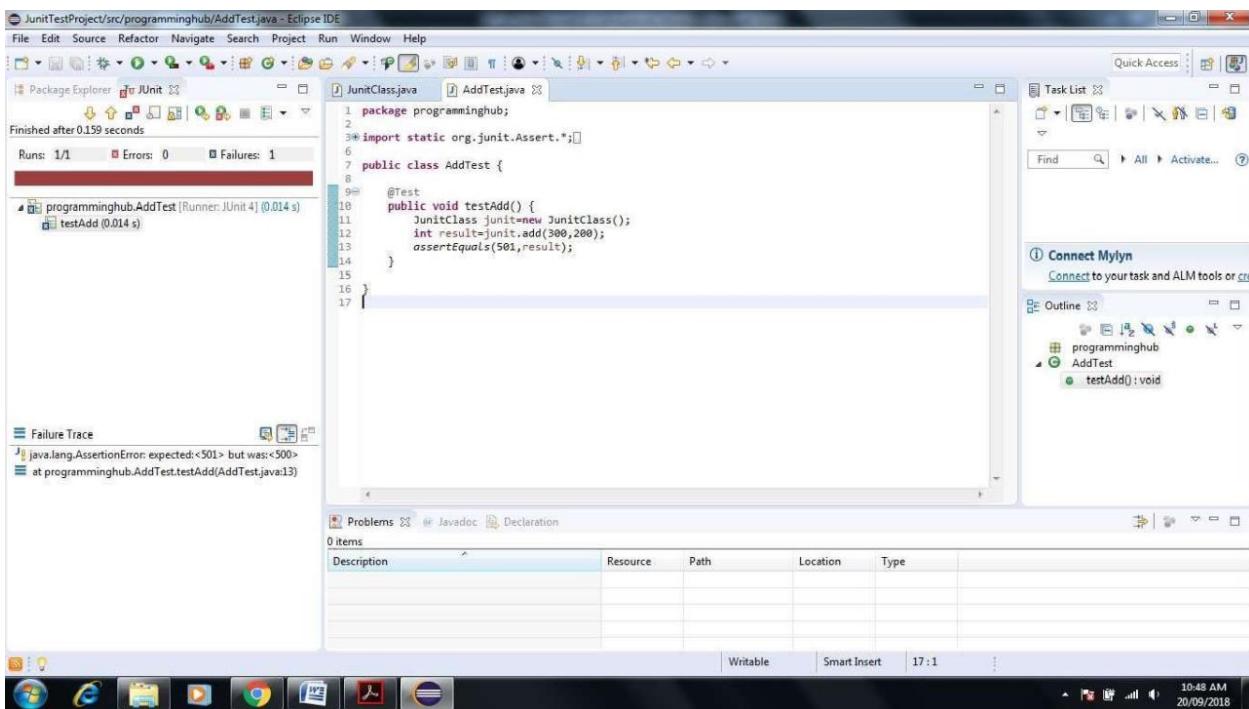
21. Let us purposely give wrong input in assertEquals method or unexpected result here we write 501 instead of 500 indicate wrong addition result

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** JunitTestProject/src/programminghub/AddTest.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbars:** Standard toolbar with icons for file operations.
- Left Sidebar:** Package Explorer (selected), JUnit (disabled).
- Middle Area:** Editor showing AddTest.java code:

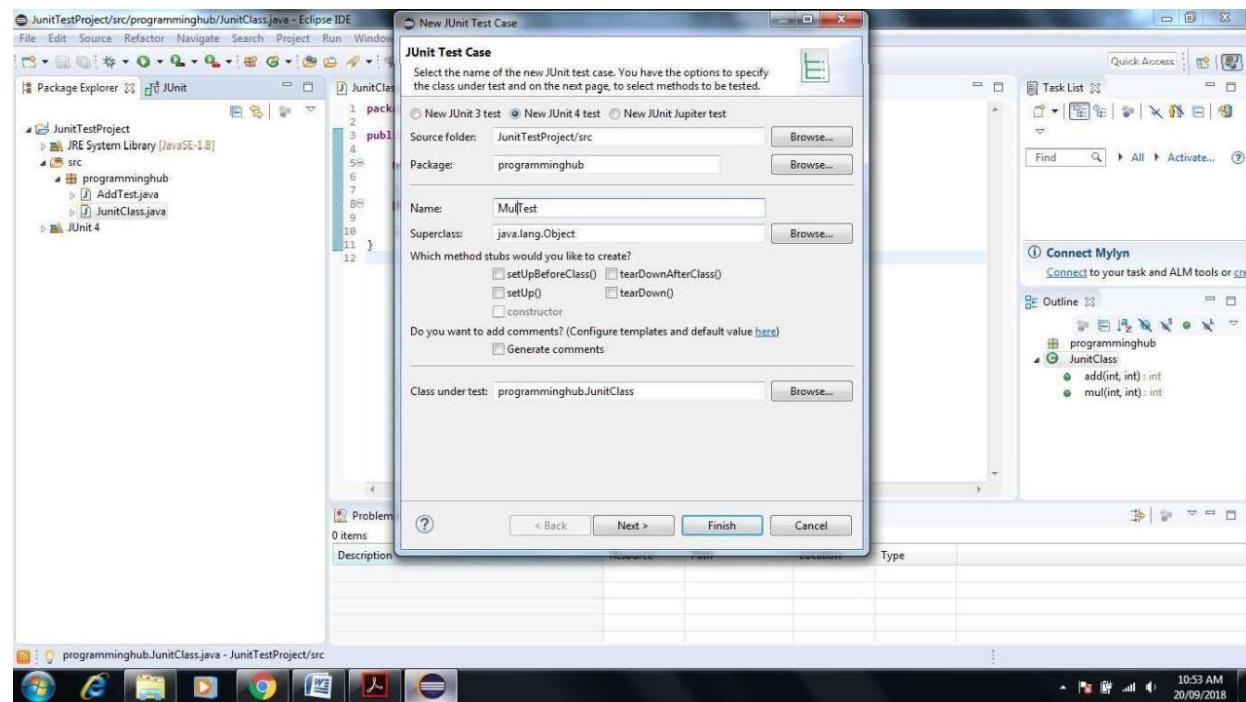
```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class AddTest {
6
7     @Test
8     public void testAdd() {
9         JUnitClass junit=new JUnitClass();
10        int result=junit.add(300,200);
11        assertEquals(501,result);
12    }
13
14 }
15
16 }
```
- Bottom Status Bar:** 10:46 AM, 20/09/2018

22. Now test case should fail.(Brown Color Bar Indicate) So again run Add Test as follow

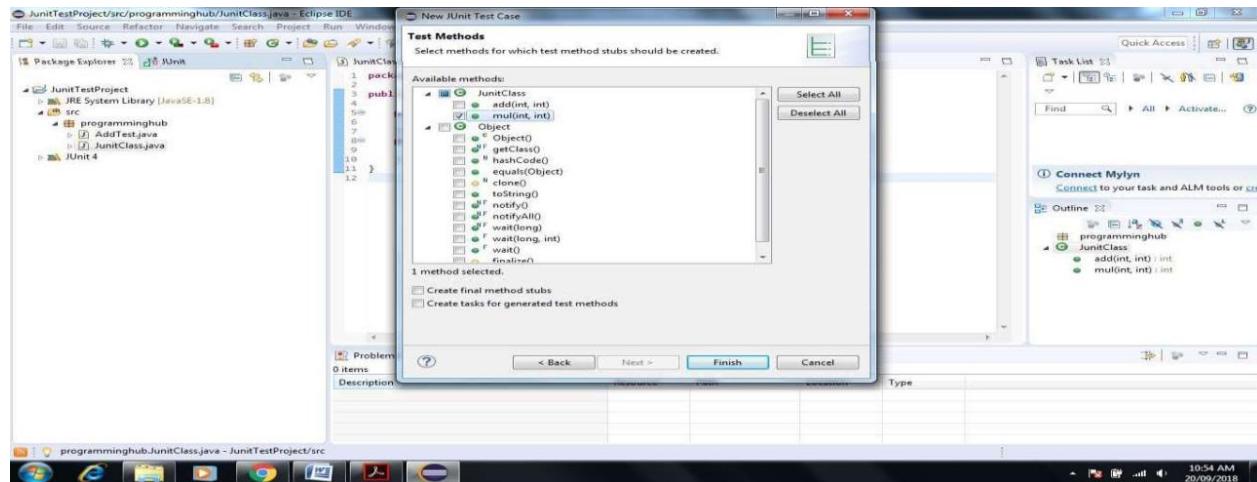


23. Similarly you can Create Test case for Multiplication Function

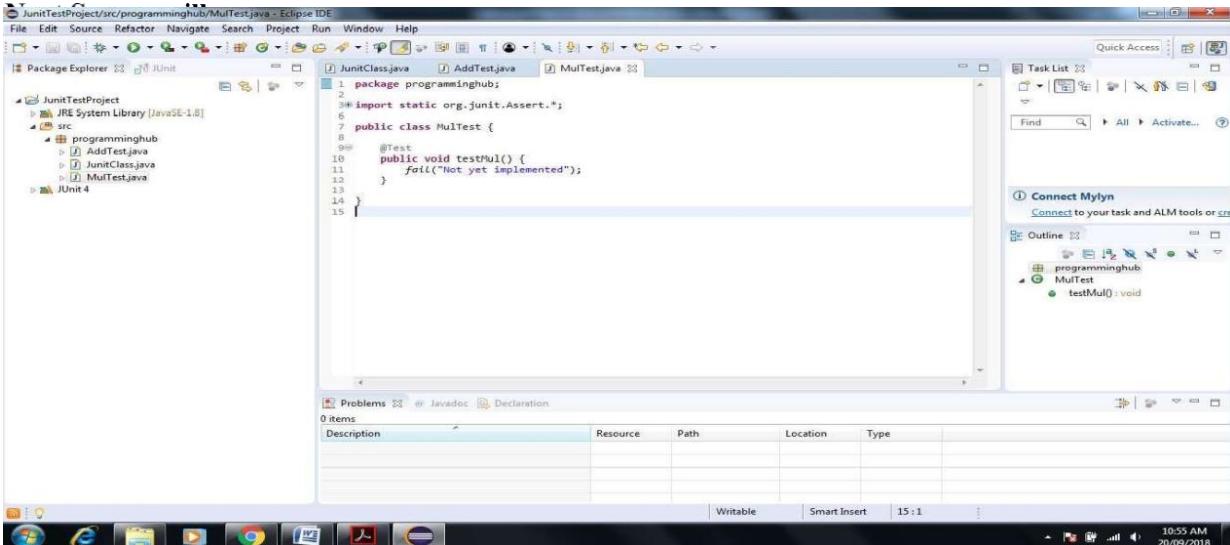
Click on Project Explorer Screen-> Right Click on JunitClass->New->JUnit Test Case-> Give name MulTest



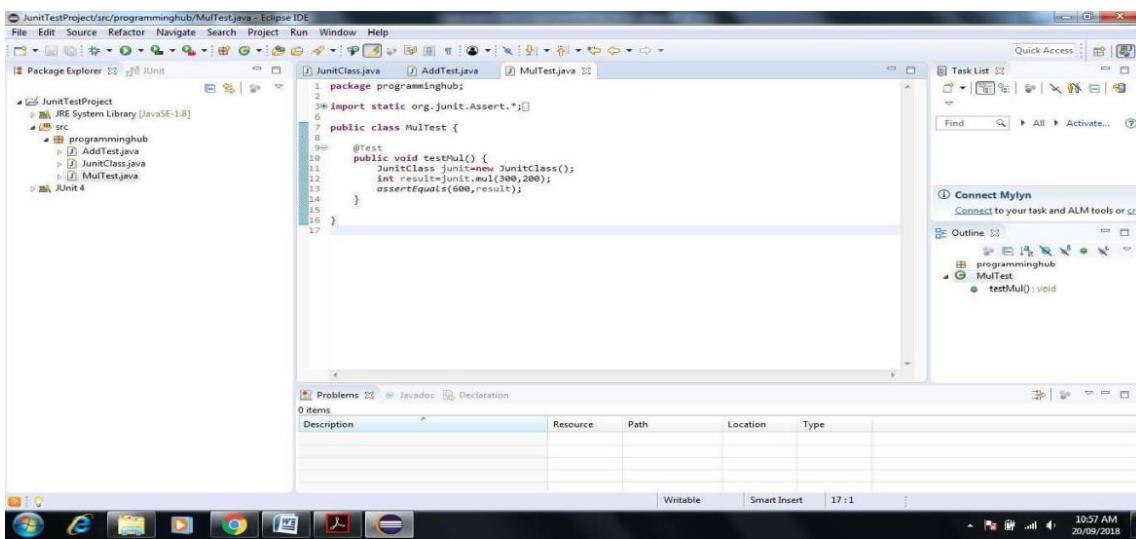
24. Click on Next->Select Mul Check Box -> Click Finish



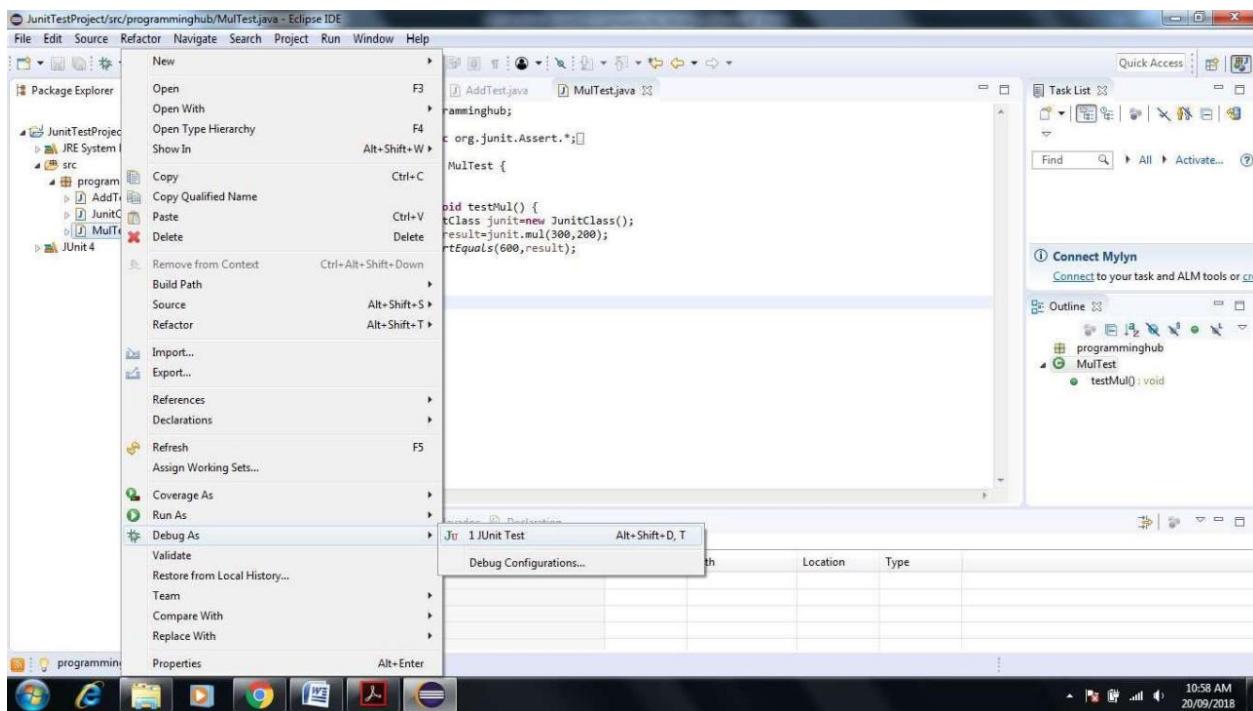
25.



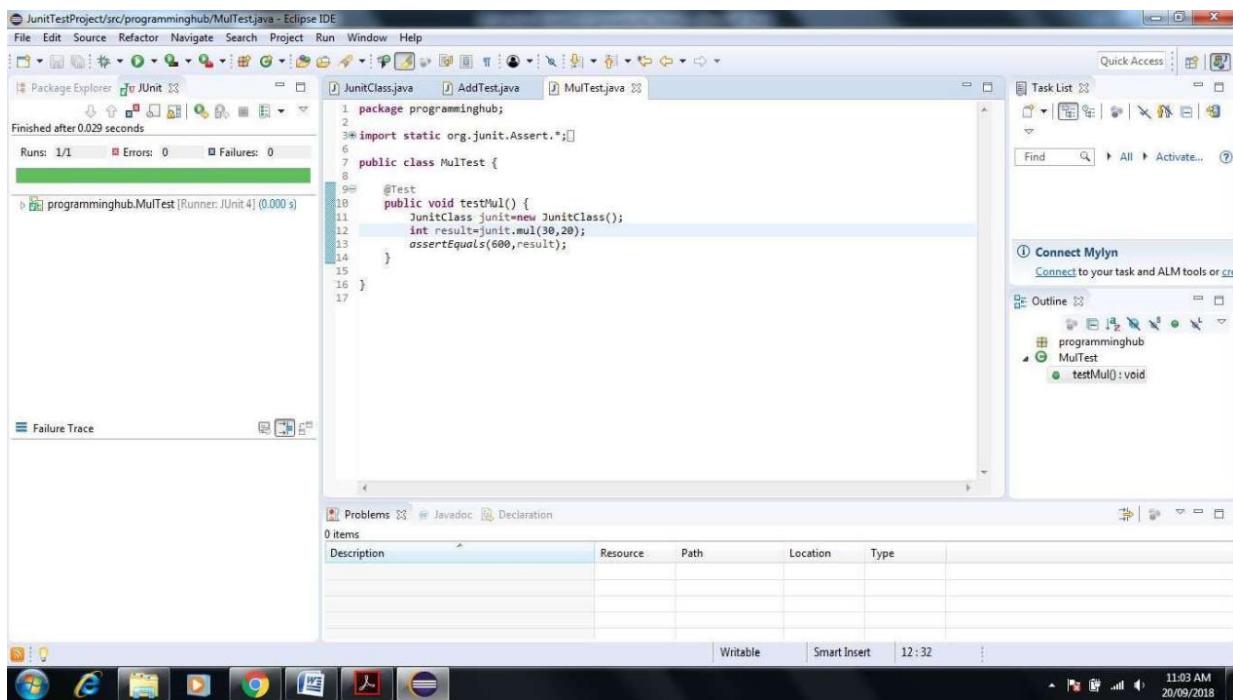
26. Write a Test Case Code inside MulTest method



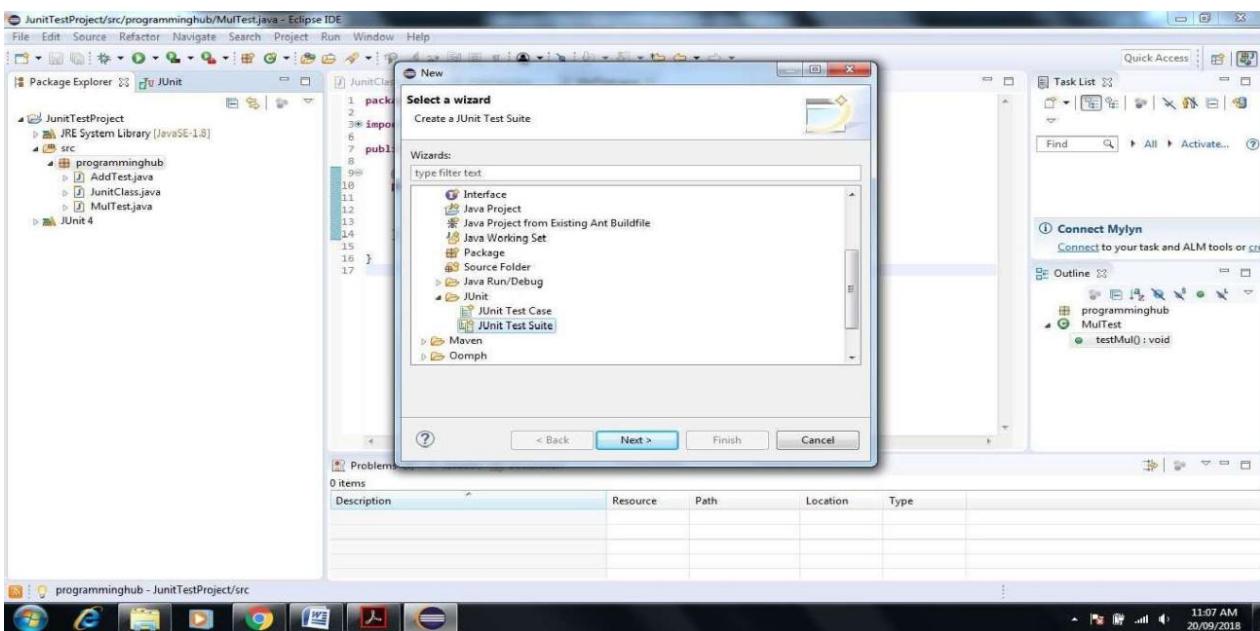
27. Right Click on MuTest->Debug->JUnit Test



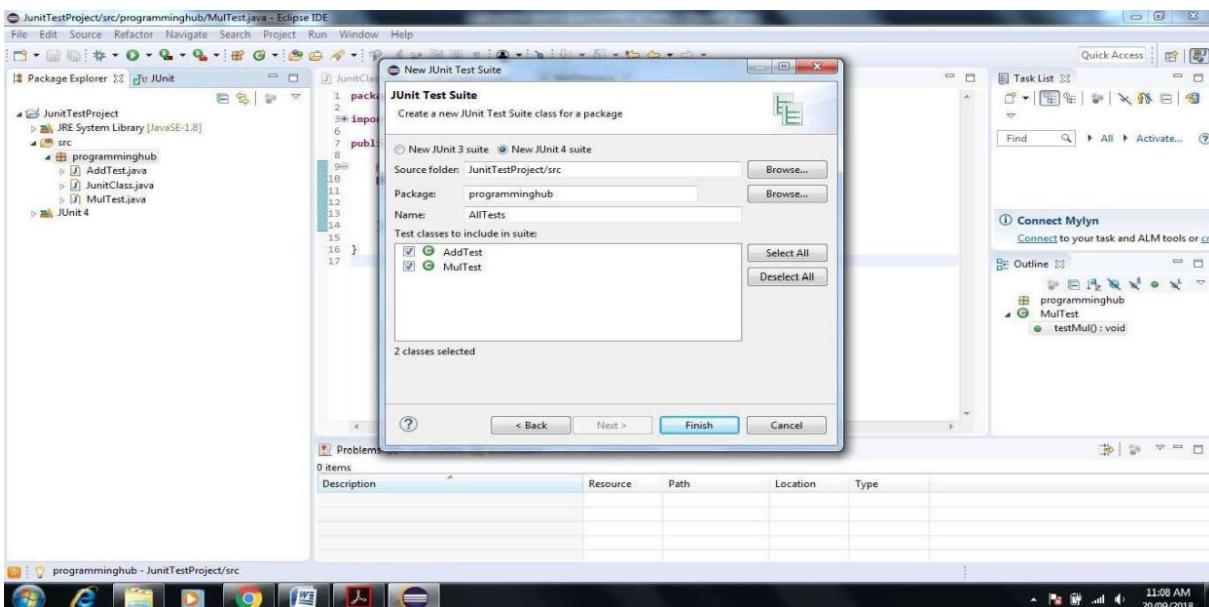
28. Execute Test



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



30. Click on Finsh



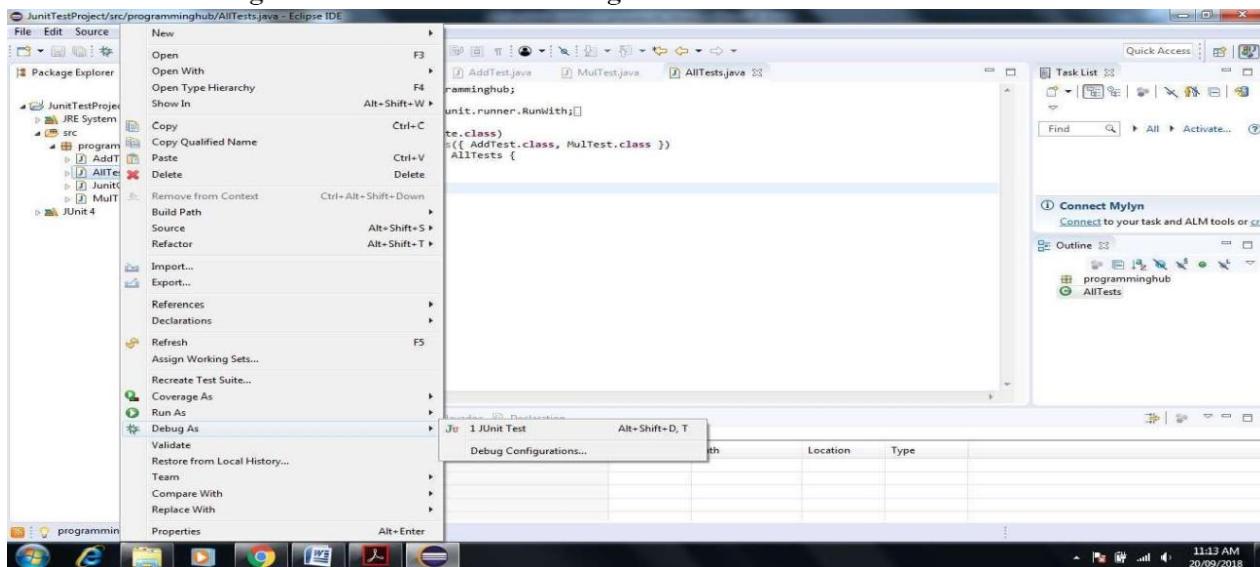
31. Next Screen Appear that automatically create Test Suite for Add and Mul

The screenshot shows the Eclipse IDE interface with the title "JUnitTestProject/src/programminghub/AllTests.java - Eclipse IDE". The code editor displays the following Java code:

```
1 package programminghub;
2
3 import org.junit.runner.RunWith;
4
5 @RunWith(Suite.class)
6 @SuiteClasses({ AddTest.class, MulTest.class })
7 public class AllTests {
8
9 }
```

The Package Explorer view on the left shows a project named "JunitTestProject" with a "src" folder containing "AddTest.java", "AllTests.java", "JunitClass.java", and "MulTest.java". The JUnit view indicates "AllTests" is selected.

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



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

The screenshot shows the Eclipse IDE interface with the title "JUnitTestProject/src/programminghub/AllTests.java - Eclipse IDE". The status bar at the bottom left shows "Runs: 2/2 Errors: 0 Failures: 0" and "Finished after 0.031 seconds". The code editor shows the same Java code as in the previous screenshot. The status bar at the bottom right shows the time as 11:14 AM and the date as 20/09/2018.