CSCB07 – Software Design

Lab 1

Objective

Getting familiar with SVN and Eclipse IDE.

Logistics

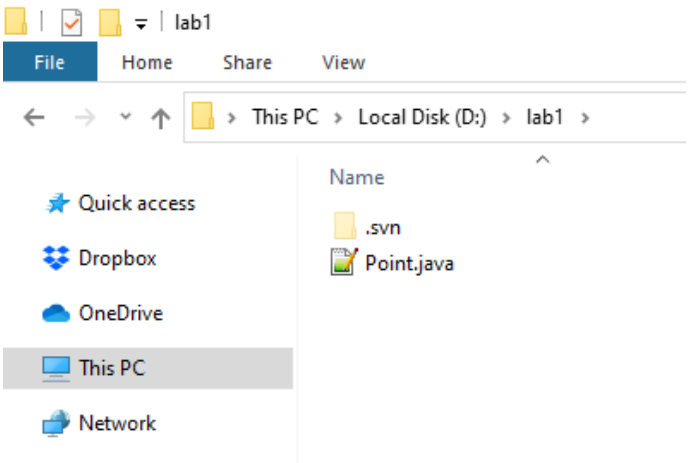
- This lab will be supervised by your TA during the tutorial session of Week 2 (May 16-20, 2022). If you encounter any problem while doing the steps listed in the next section, ask the TA for help.
- Make sure that SVN and Eclipse are installed on your machine as per the instructions posted on Quercus.
- The lab should be done individually.

Instructions

1. Open Terminal/Command Prompt and type the following command where X is your tutorial's section number (1, 2, 3, or 4):

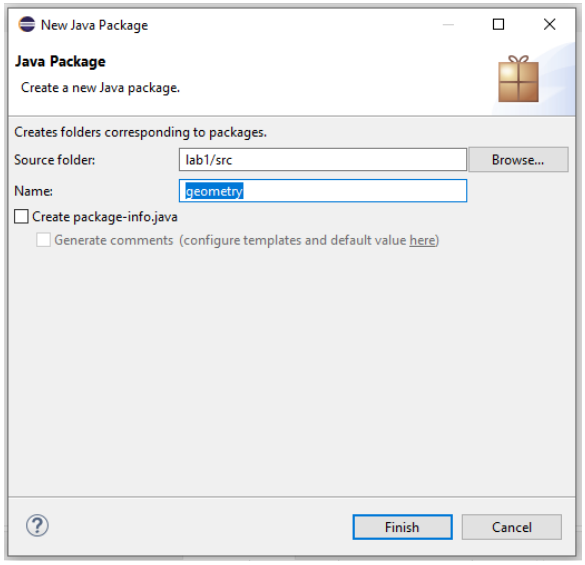
   svn checkout -r 1 https://helixteamhub.cloud/UTSC/projects/tutX/repositories/subversion/lab1

   When prompted for credentials use cscb07 and Cscb07#s22 for username and password respectively. A working copy of the repository will be created in the directory where you run the command:

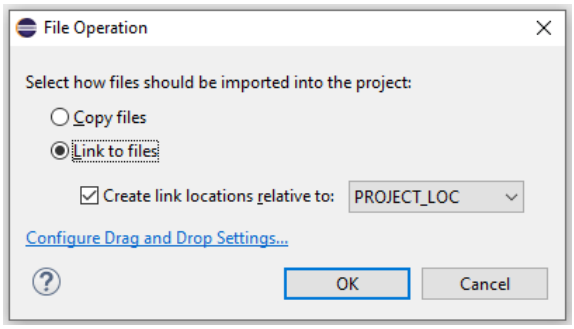


2. Open Eclipse IDE and create a new project:
   "File" -> "New" -> "Java Project" -> Name it "lab1" and click "Finish" (don't create a module for now)

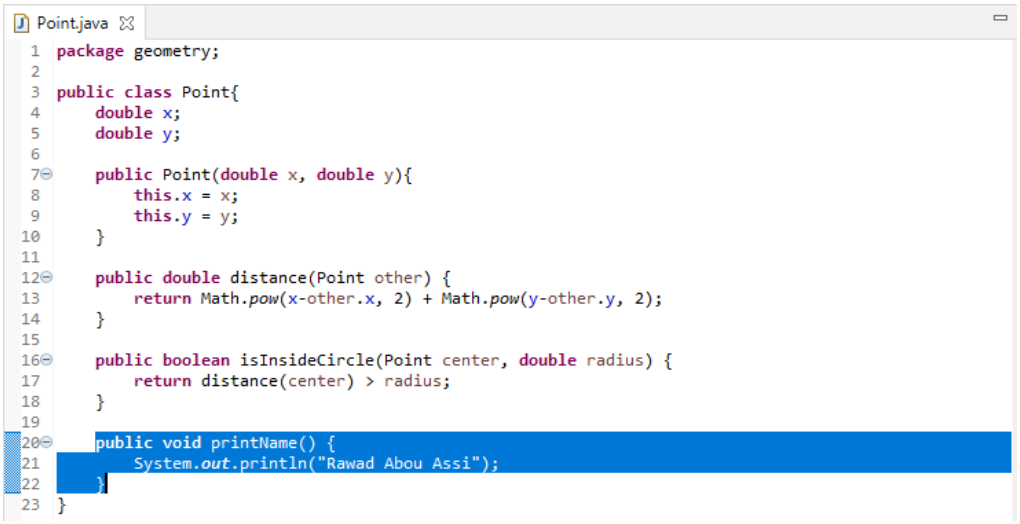
3. Create a package and name it “geometry”:
   Right-click “lab1” in the Package Explorer -> “New” -> “Package” -> Name it “geometry” and click “Finish”



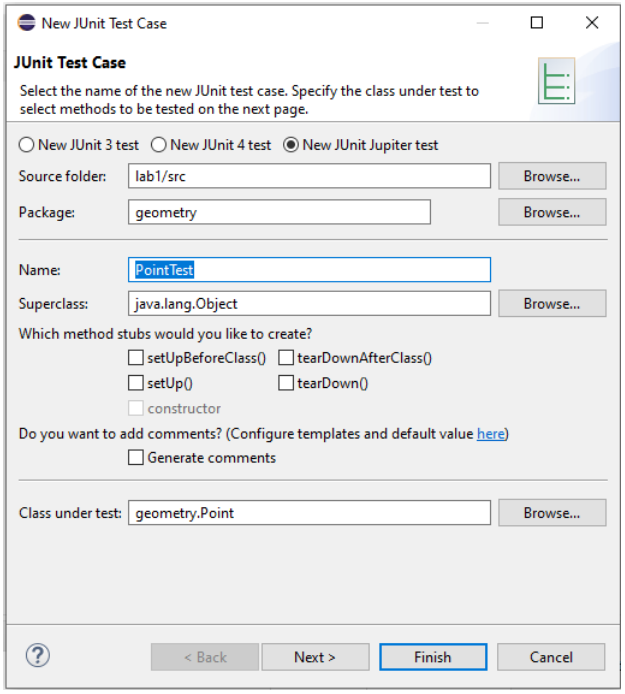
4. Drag-and-drop Point.java from the working copy into the geometry package and make sure to select “Link to files” so that any changes made in Eclipse would be reflected in the working copy:



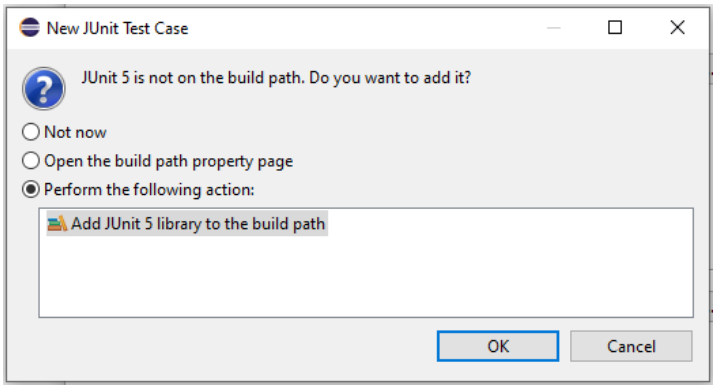
5. Add a one-line method to Point.java that prints your name. The class should look as follows afterward:



6. Add a test class for Point.java:
   Right-click “Point.java” in the package explorer -> “New” -> “JUnit Test Case” -> “Finish”



   When prompted, add JUnit 5 library to the build path



7. In PointTest.java, delete the following:

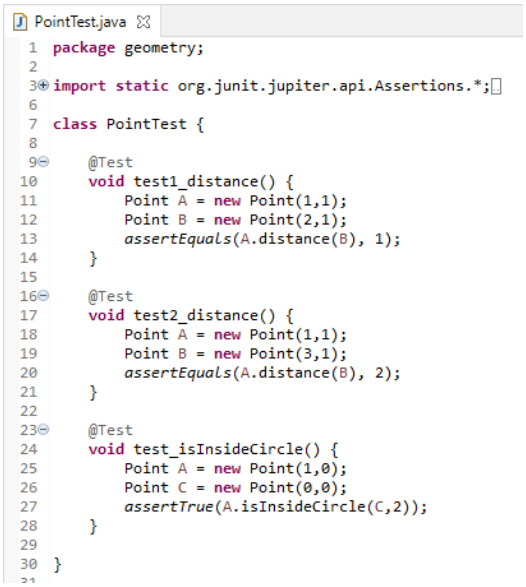
```
@Test
void test() {
	fail("Not yet implemented");
}
```

8. And add the following instead:

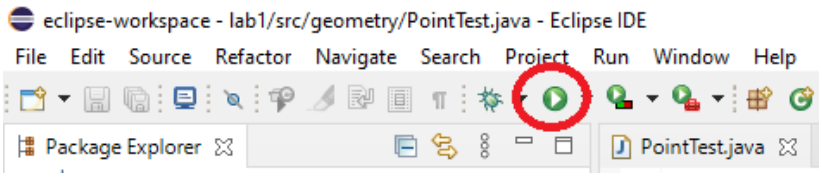
```
@Test
void test1_distance() {
	Point A = new Point(1,1);
```

```
        Point B = new Point(2,1);
        assertEquals(A.distance(B), 1);
    }

    @Test
    void test2_distance() {
        Point A = new Point(1,1);
        Point B = new Point(3,1);
        assertEquals(A.distance(B), 2);
    }

    @Test
    void test_isInsideCircle() {
        Point A = new Point(1,0);
        Point C = new Point(0,0);
        assertTrue(A.isInsideCircle(C,2));
    }
```

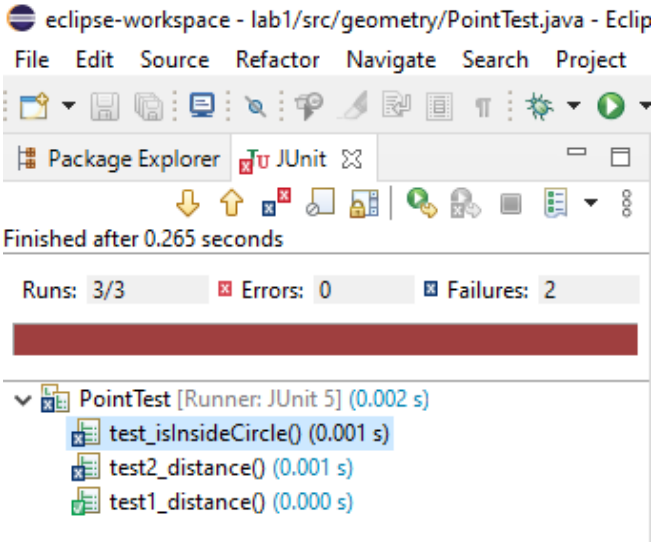
9. After steps 7 and 8, PointTest.java should look as follows:



10. Run the tests in PointTest.java



11. Two of the tests will fail indicating potential bugs in Point.java

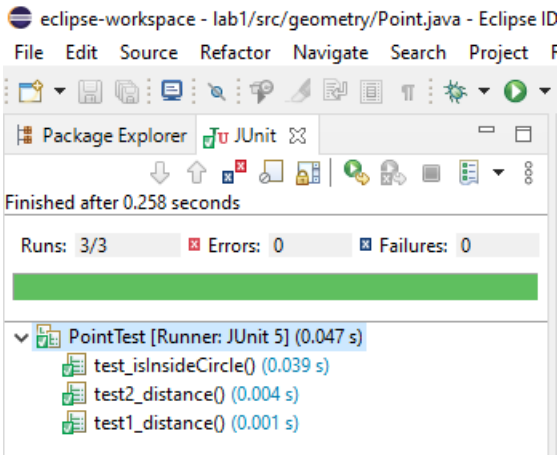


12. Go to Point.java and fix the bugs:

    Fixing bug 1
    return Math.sqrt(Math.pow(x-other.x, 2) + Math.pow(y-other.y, 2));

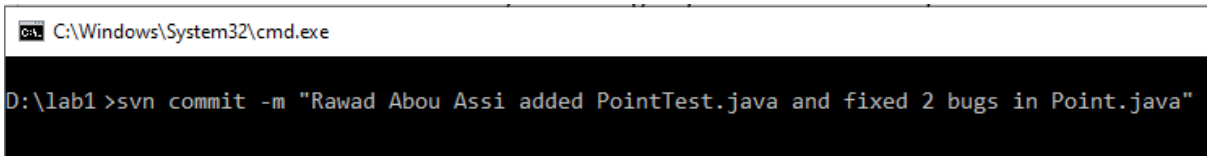
    Fixing bug 2
    return distance(center) < radius;

13. Re-run the tests. All of them should pass now



14. As mentioned earlier, the changes you make on Point.java will be reflected in the working directory. As for the newly created PointTest.java, we need to copy it there as follows:
    a. Right-click PointTest.java in the Package Explorer -> “Properties”
    b. Go the location you obtain in step a, copy PointTest.java and paste it in the working directory
    c. Open a terminal in the working directory and type: **svn add PointTest.java**

15. Commit all the changes and make sure to add your name in the message



16. If you get an out-of-date error in step 15, do the following:
    a. Type: **svn update**
    b. Type m to merge: **m**
    c. Type 2 to select your version: **2**
    d. Type r to mark as resolved: **r**
    e. Commit by using the same command of step 15