

Object Oriented Programming

Week 1 Part 2

Setting up Eclipse to Create a Class

Setting up a Java Object using Eclipse

Eclipse

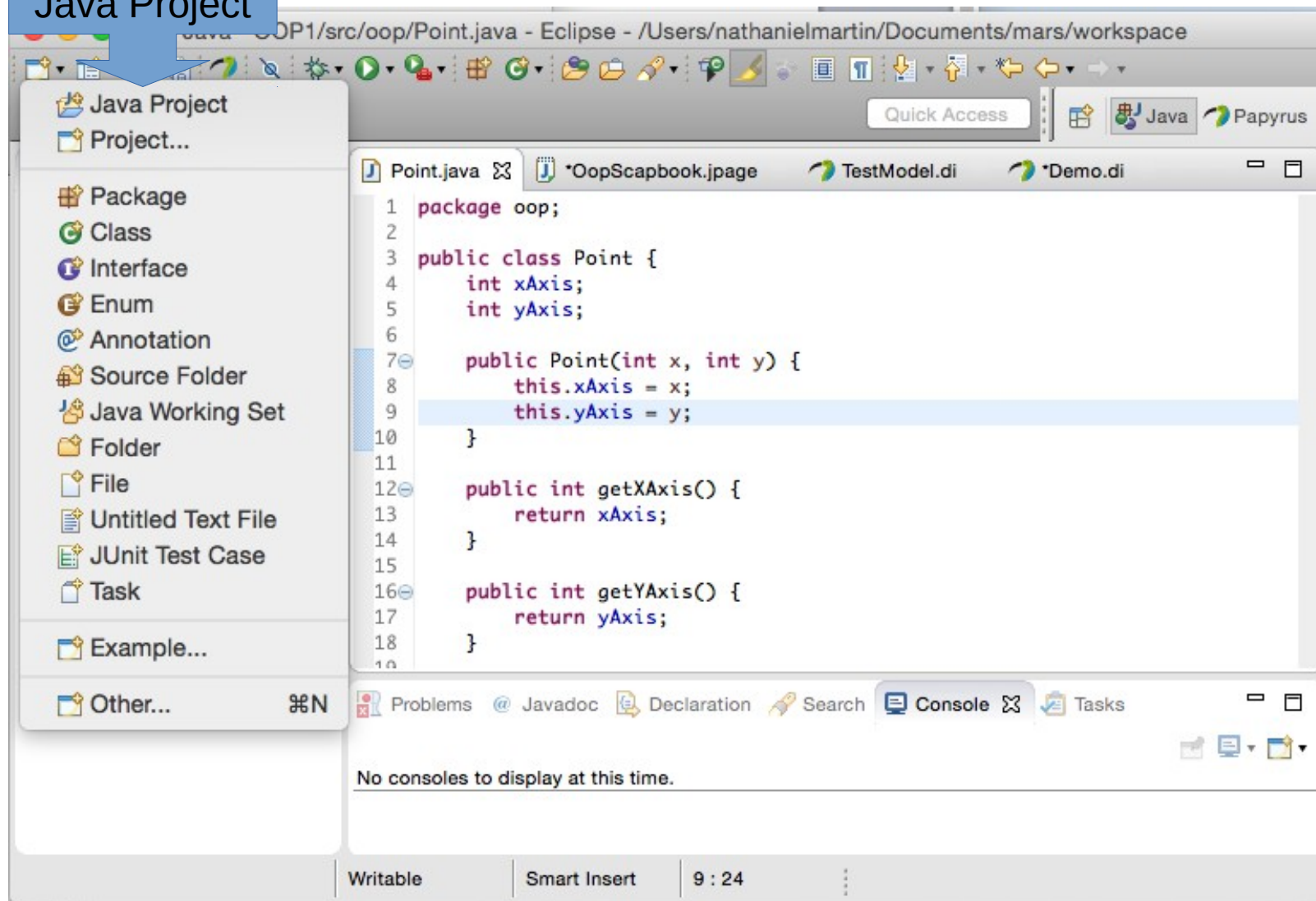
- Eclipse is an IDE (Integrated Development Environment)
 - It provides an editor
 - A build system and compiler
 - A test system
 - A Software Configuration System
 - And more
- IDEs simplify programming, once you learn to use them.

Writing a Program in Eclipse

1. Create a Project
2. Inside the project create a class
 - Add the class to a package
3. Add the instance variables and methods to the class.
 - The IDE notifies you immediately if there are compiler errors

Create a Project

Add a new
Java Project



Name your Project

The screenshot shows the 'New Java Project' dialog box. A blue callout box with the text 'Give it a name' and a downward arrow points to the 'Project name' text field, which contains the word 'Demo'. Another blue callout box with the text 'Click Finish' and a downward arrow points to the 'Finish' button at the bottom right of the dialog. The dialog includes sections for 'Location', 'JRE', 'Project layout', and 'Working sets'. The 'Project name' field is the first and most prominent input field.

Give it a name

Project name: Demo

☒ Use default location

Location: /Users/nathanielmartin/Documents/mars/workspace/Demo [Browse...](#)

JRE

☒ Use an execution environment JRE: JavaSE-1.8

☐ Use a project specific JRE: Java SE 8 [1.8.0_05]

☐ Use default JRE (currently 'Java SE 8 [1.8.0_05]') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets

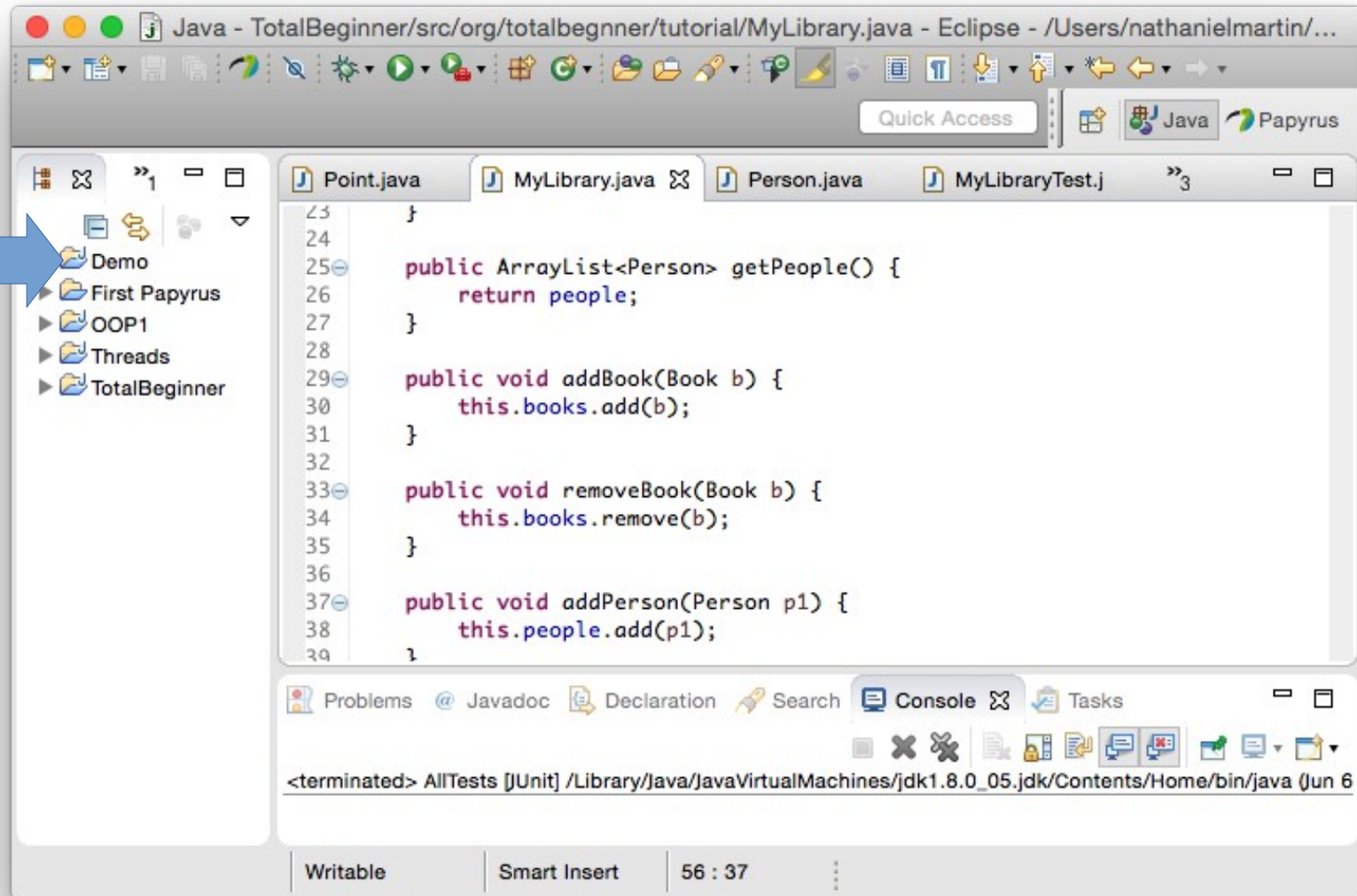
Working sets: [Select...](#)

Click Finish

[? < Back](#) [Next >](#) [Cancel](#) [Finish](#)

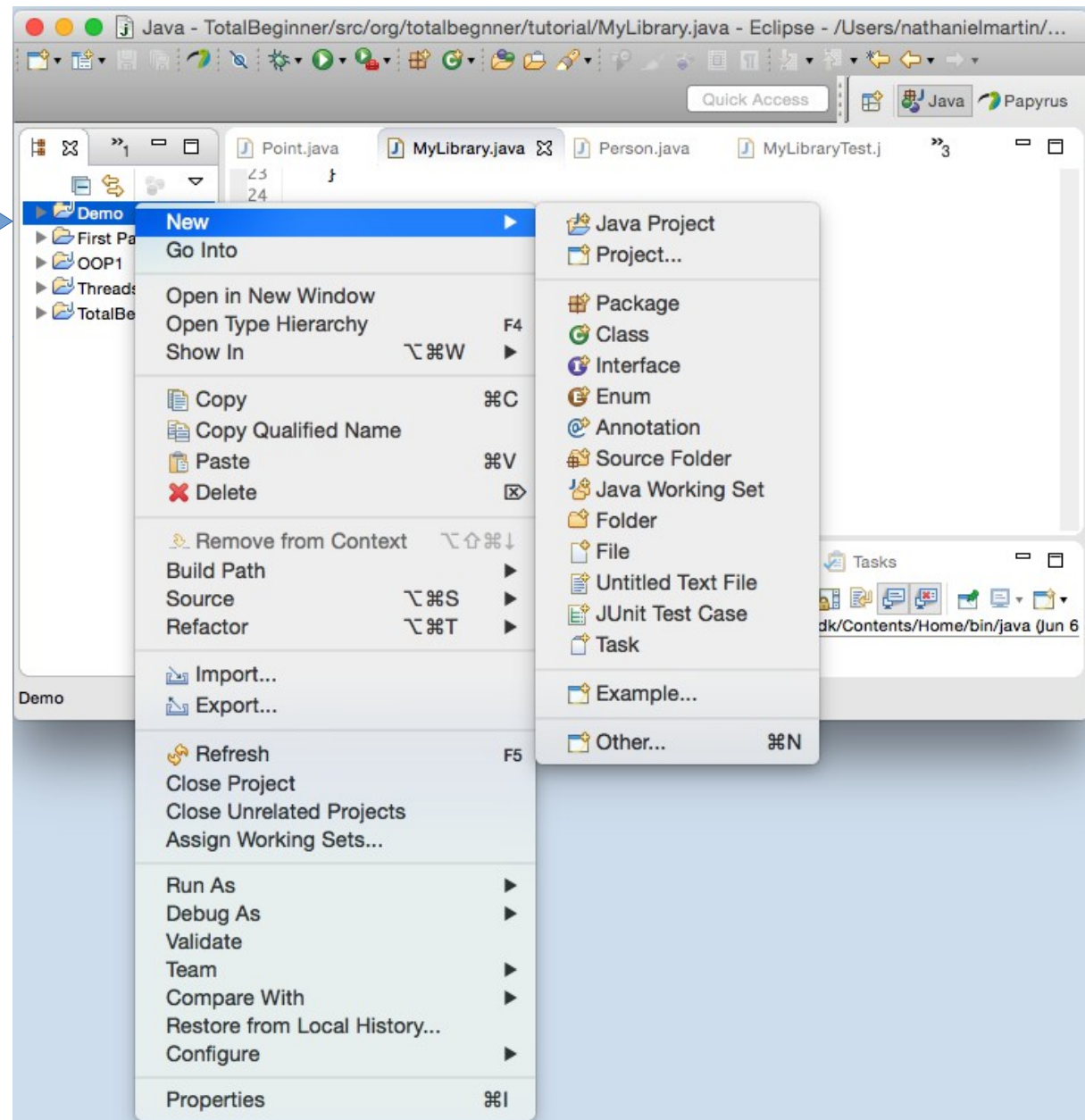
Now you have a project

Demo
project

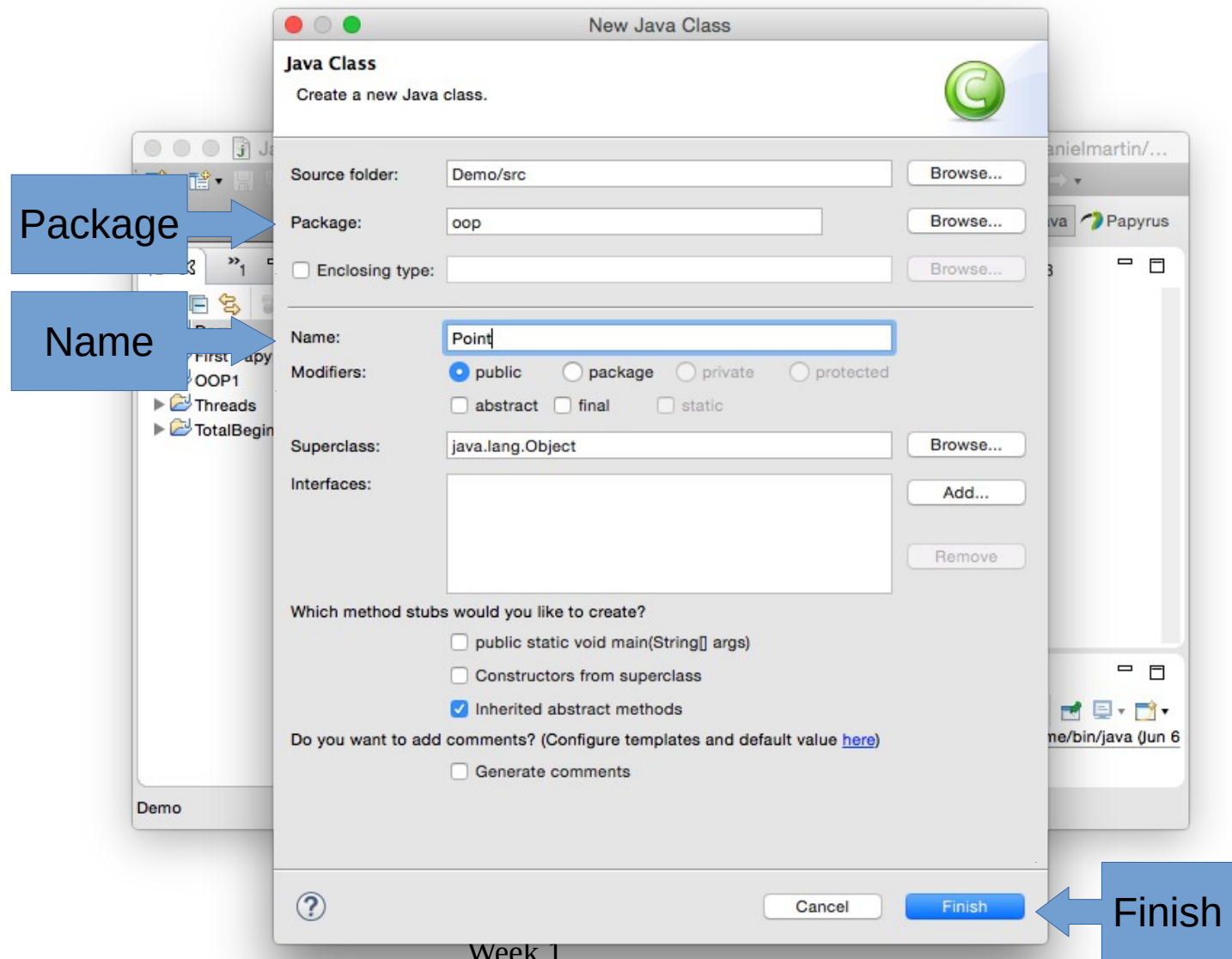


Add a Class

1. Right click project
2. Roll over New
3. Select Class

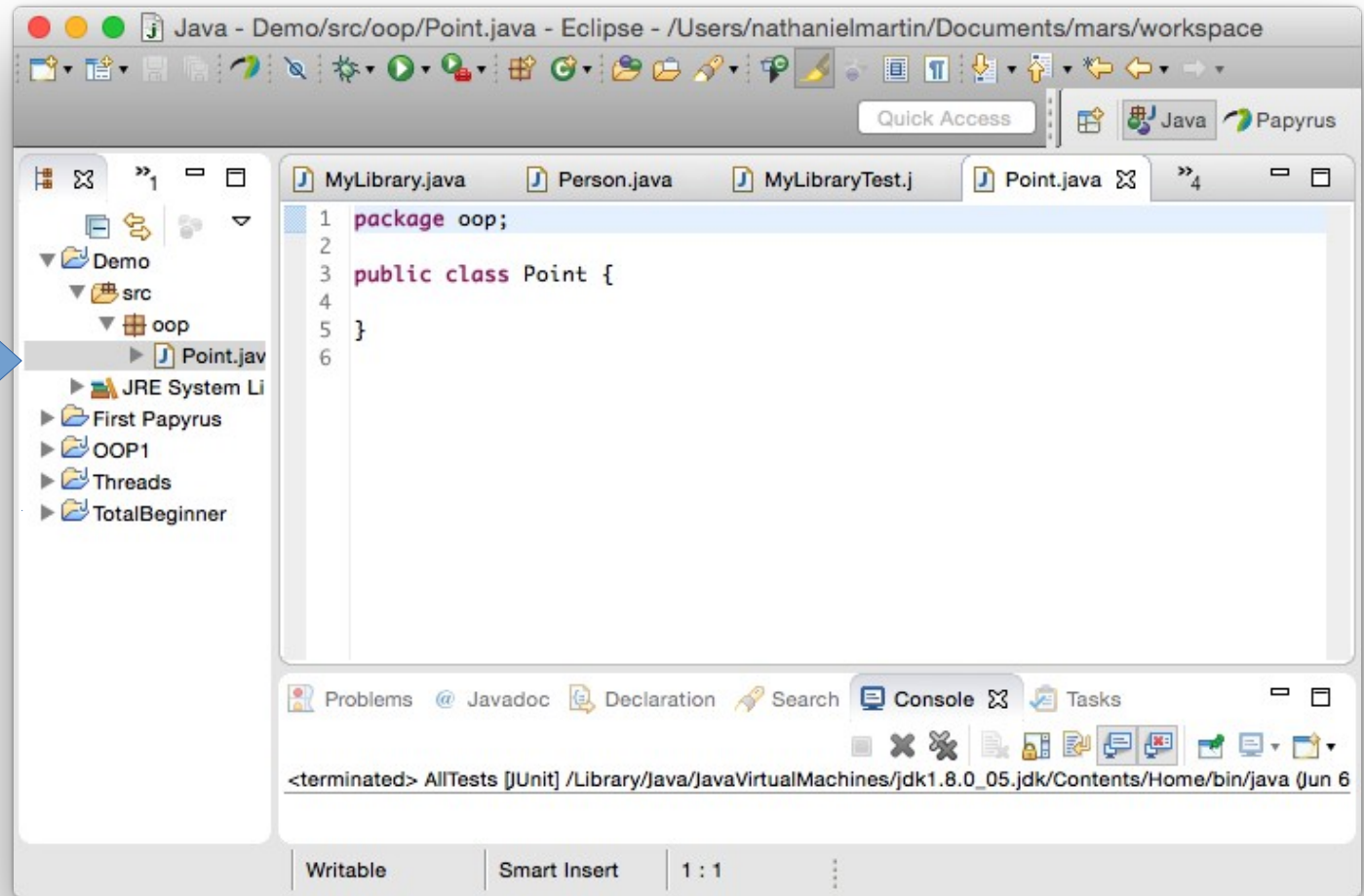


Add Class Name and Package



We have a class

Class **Point** in
Folder **src** and
Package **oop**



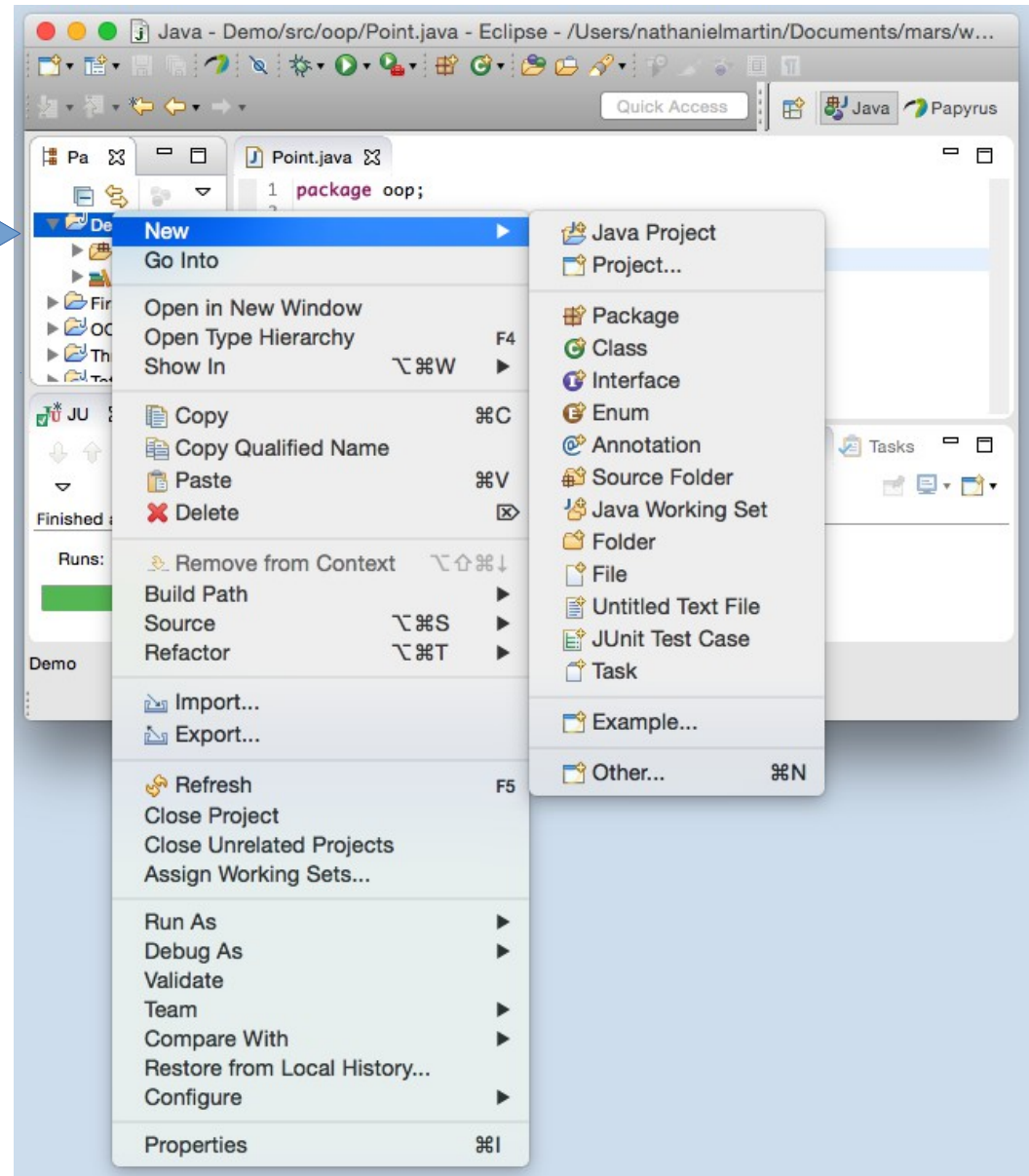
Setting up Eclipse to do Test Driven Development

Eclipse and TDD

- Eclipse for Java is designed for TDD
 - JUnit is integrated into the Java development environment
 - You can run JUnit tests from the IDE
 - You can access failing tests with a click of the mouse
 - Eclipse suggests solutions to failing tests
- But first you need to set up the tests

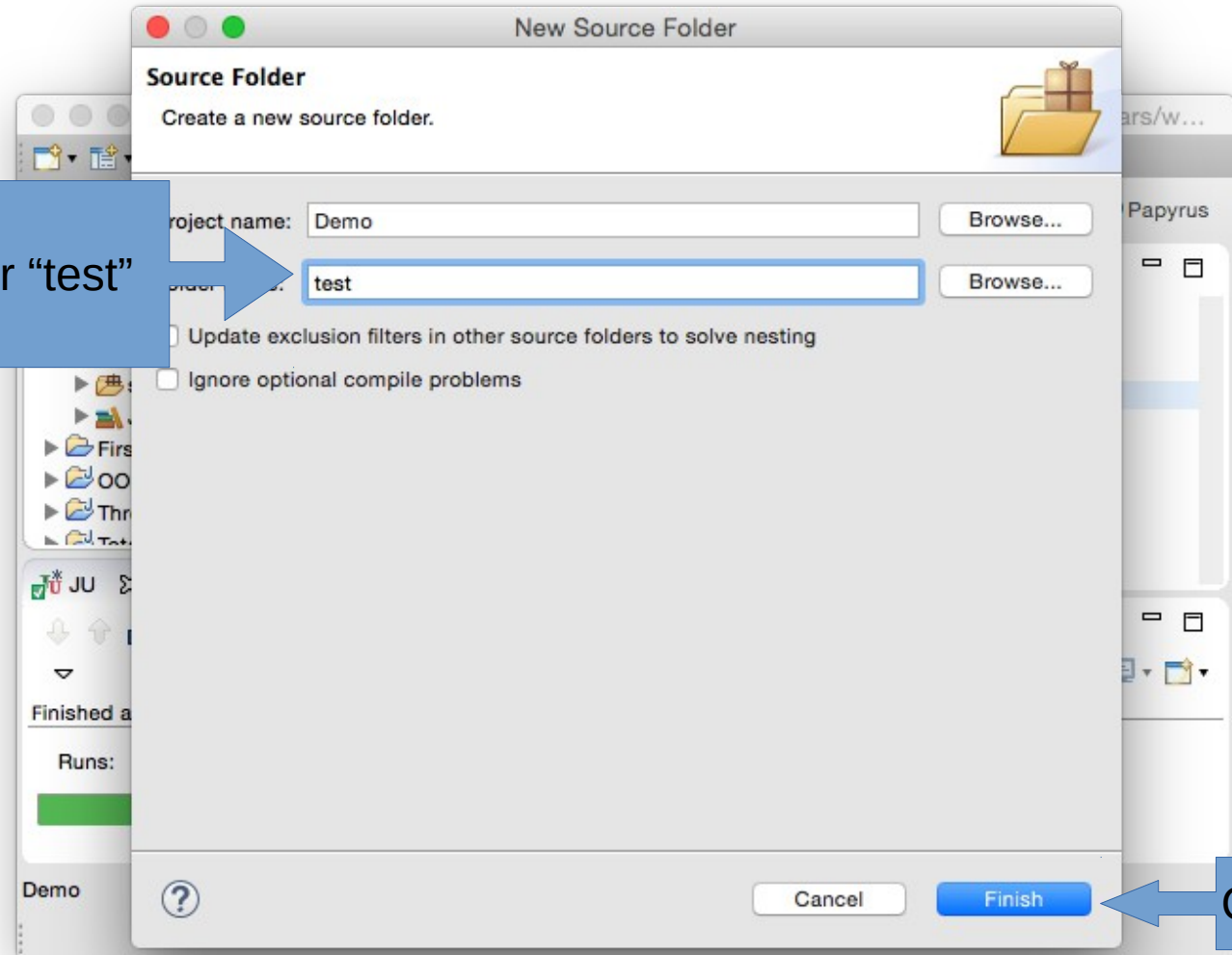
New Source Folder for Tests

1. Right Click Project
2. Select New
3. Select Source Folder



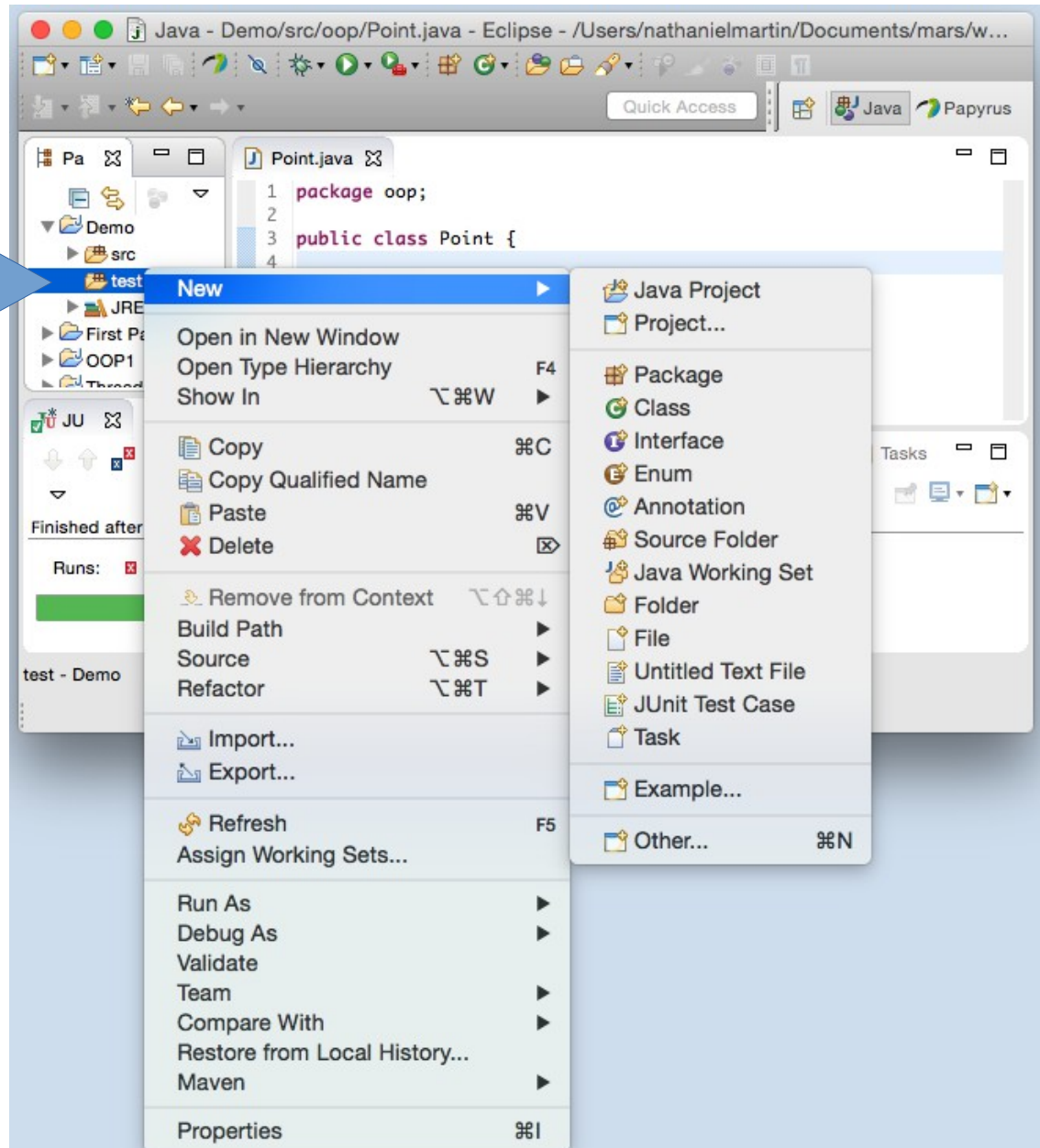
Give the folder a name

Name the new folder "test"



Make a New Junit Test

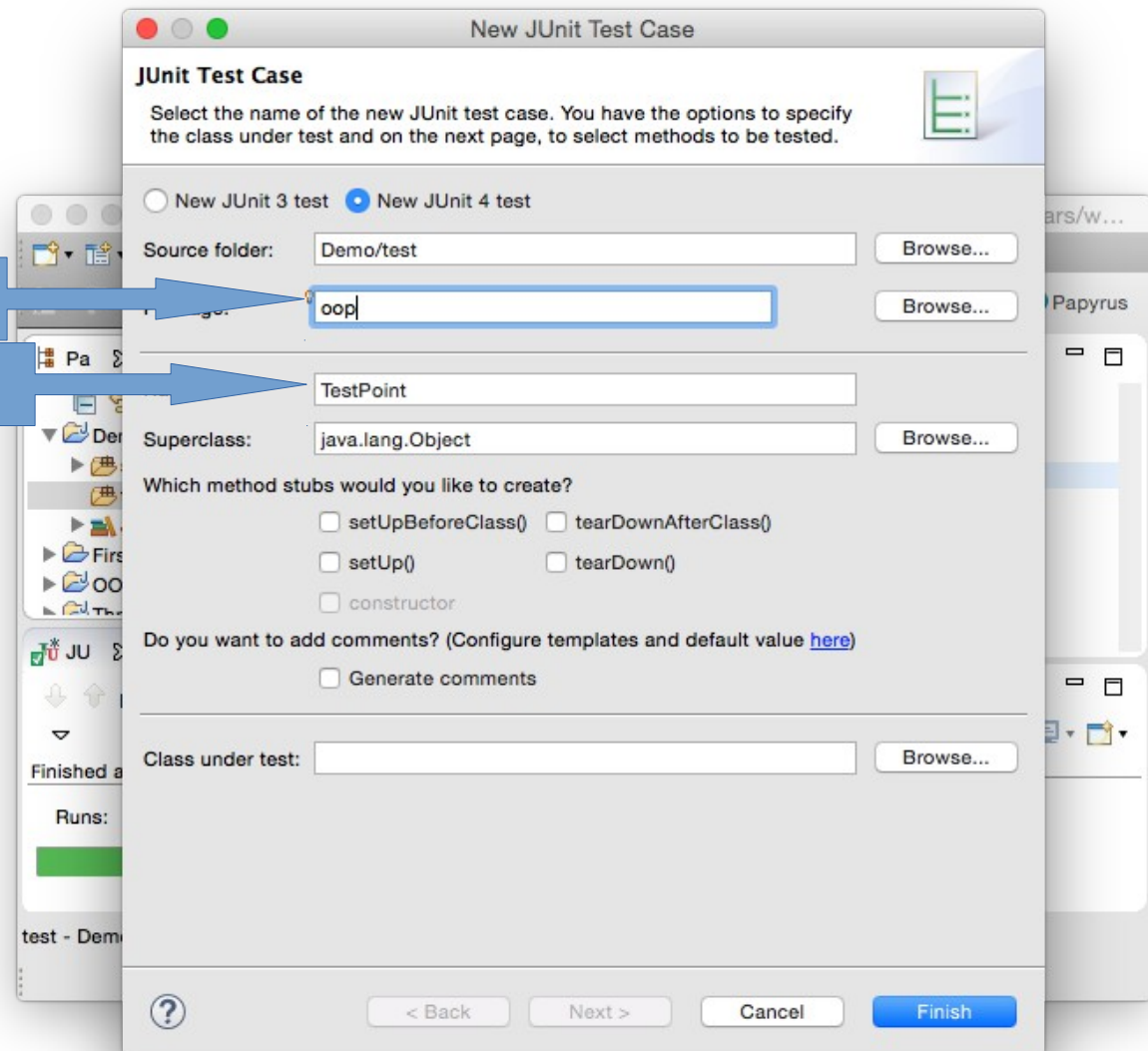
1. Right click new folder
2. Select New
3. Select Junit Test Case



Name the test and add it to package

Put it in the “oop” package

Name it “TestPoint”



Add JUnit 4 to the build path

