EECS 3311
Software Design
Lab 0 (0 points), Version 1

Instructor: Song Wang
Release Date: Jan 13, 2020

There is no submission required for this Lab 0, but you should complete it as soon as possible. All your lab submissions must be compilable on the department machines. It is then crucial that should you choose to work on your own machine, you are responsible for testing your project before submitting it for grading.

This lab is intended to help you get familiar with the programming/design environment of this course. We will learn about the most common features of Eclipse for Java.

Check the Amendments section of this document regularly for changes, fixes, and clarifications.

Ask questions on the course forum on the eClass site.

Policies

Your (submitted or un-submitted) solution to this assignment (which is not revealed to the public) remains the property of the EECS department. Do not distribute or share your code in any public media (e.g., a non-private Github repository) in any way, shape, or form before you get the permission from your instructors.

- You are required to **work on your own for this lab**. No group partners are allowed.

- When you submit your solution, you claim that it is solely your work. Therefore, it is considered as an violation of academic integrity if you copy or share any parts of your code or documentation.

- When assessing your submission, the instructor and TA may examine your doc/code, and suspicious submissions will be reported to the department/faculty if necessary. We do not tolerate academic dishonesty, so please obey this policy strictly.

- Emailing your solutions to the instruction or TAs will not be acceptable.

Submitting Your Work

> Lab 0 is a practice lab, no submission.

Amendments

- so far so good

Build Open-source Projects in Eclipse with Maven

In this lab, you are expected to learn how to build an open-source project and how to utilize Eclipse and its plugin tools to analyze the code and test cases. We will play with **Apache Commons IO**[1] in this lab. The Apache Commons IO library contains utility classes, stream implementations, file filters, file comparators, and much more.
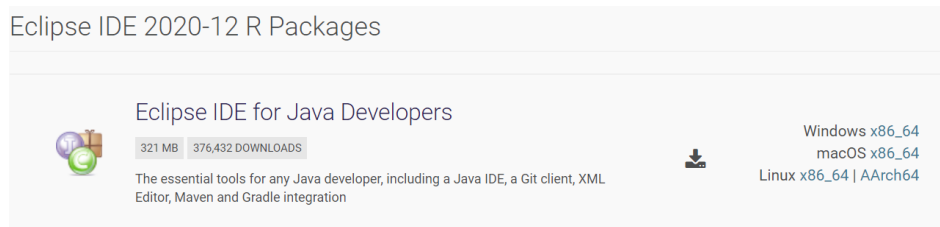
You can find the raw project of Apache Commons IO under the folder of **proj/**

0.1 Eclipse

We will use Eclipse to develop all you labs, you can download Eclipse from the following link:

https://www.eclipse.org/downloads/packages/

We will use Eclipse IDE for Java Developers for all labs/projects in this course.

The default JDK is java 8.
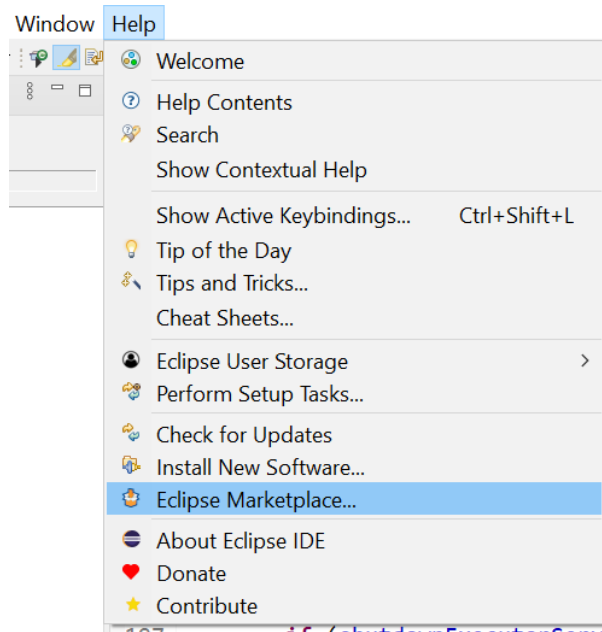
0.2 Build Projects with Maven

Figure 1: Search for Maven in Eclipse.

Maven is a build automation tool used primarily for Java projects, which is widely used by many commercial and open-source projects to help with the process of software development. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information. POM is an XML representation of a Maven project held in a file named **pom.xml**.

---

[1]https://github.com/apache/commons-io

Detail information of the structure of **pom.xml** can be found here: https://maven.apache.org/pom.html.

There is quick tutorial to help you understand how Maven works: **https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html**

Maven is a standalone tool, while Eclipse IDE provides excellent support for Maven. This tooling is developed in the M2Eclipse [2] plugin project. You can find the M2Eclipse plugin in the Eclipse Marketplace (see figure 2).

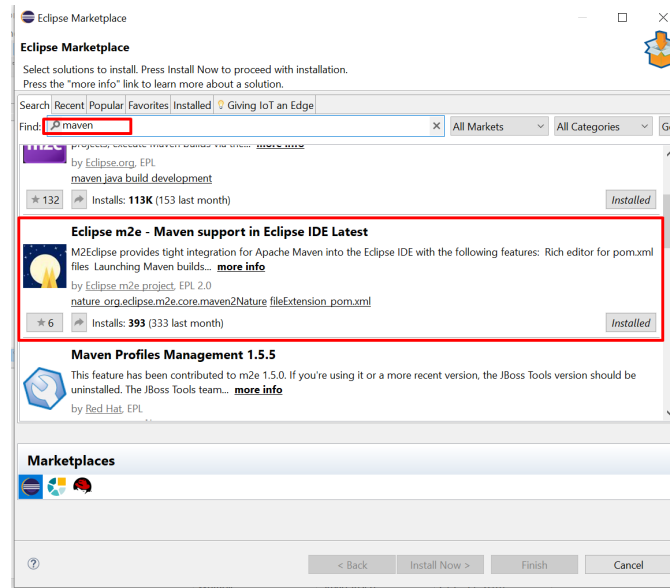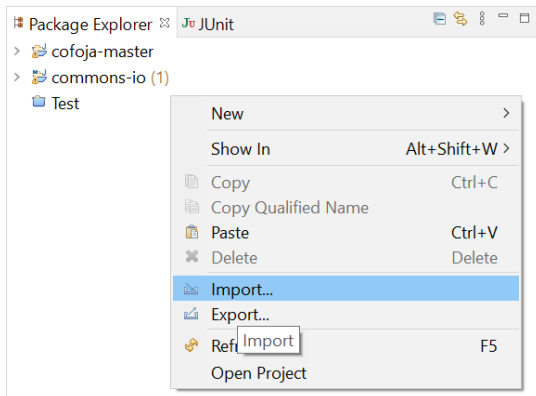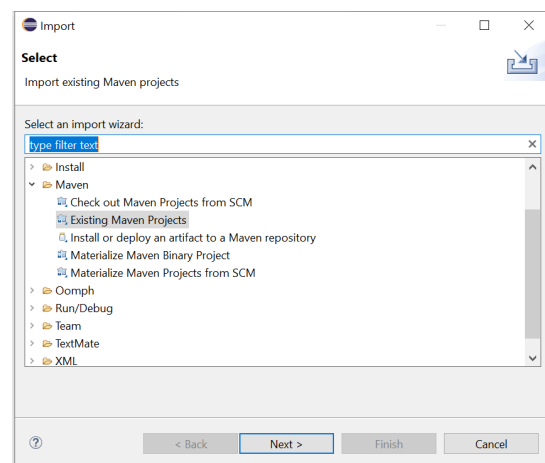You can find the M2Eclipse plugin with keyword "maven" and install the plugin.

Figure 2: Install M2Eclipse in Eclipse.

0.3 Import Maven Project into Eclipse

After the installation of Maven, your next step is to import the maven-based open source project into your Eclipse as a new maven project.

(a) Import Project Step 1.

(b) Import Project Step 2.

Figure 3: Steps to import maven projects into Eclipse.

---

[2]https://www.eclipse.org/m2e/

After you import the project into Eclipse, you will have the following structure:
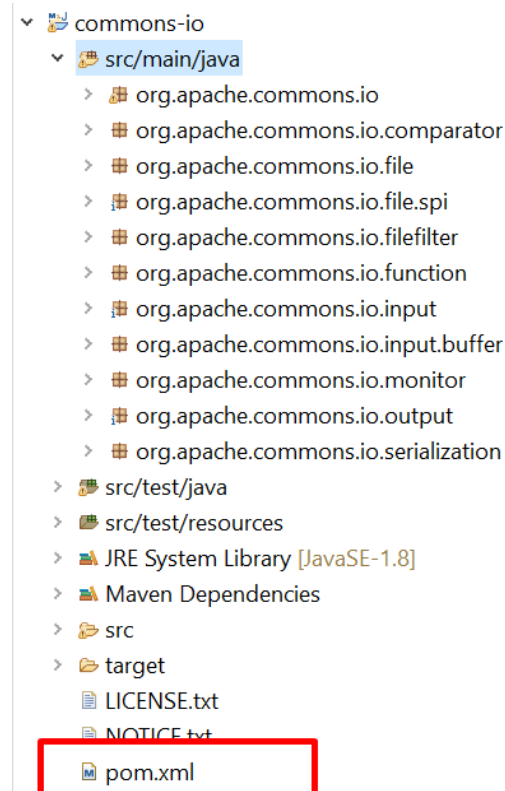

Figure 4: Structure of the imported project.

This project is maintained by Maven, you can perform many different tasks by using maven commands as follows.
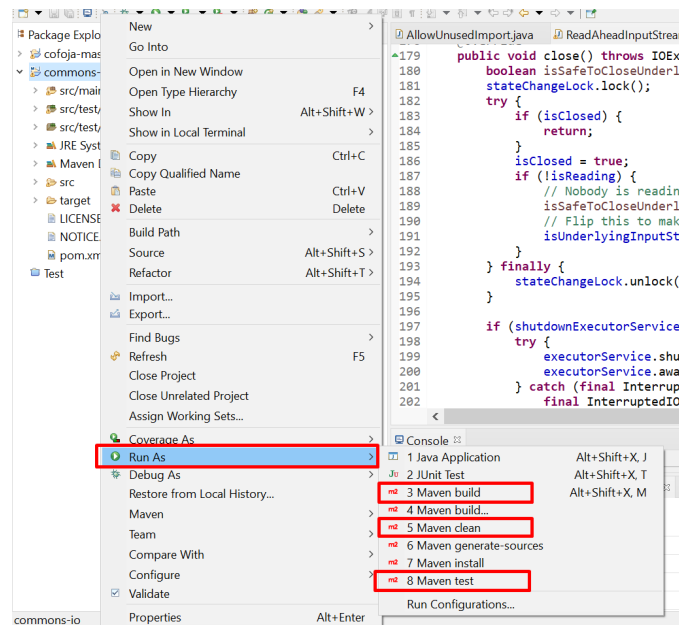

Figure 5: Maven commands.

0.4   Run Junit Test Cases

The imported project contains a list of 1,777 test cases, which is maintained by Junit. You can run specific test cases or the whole test suite with Junit commands.
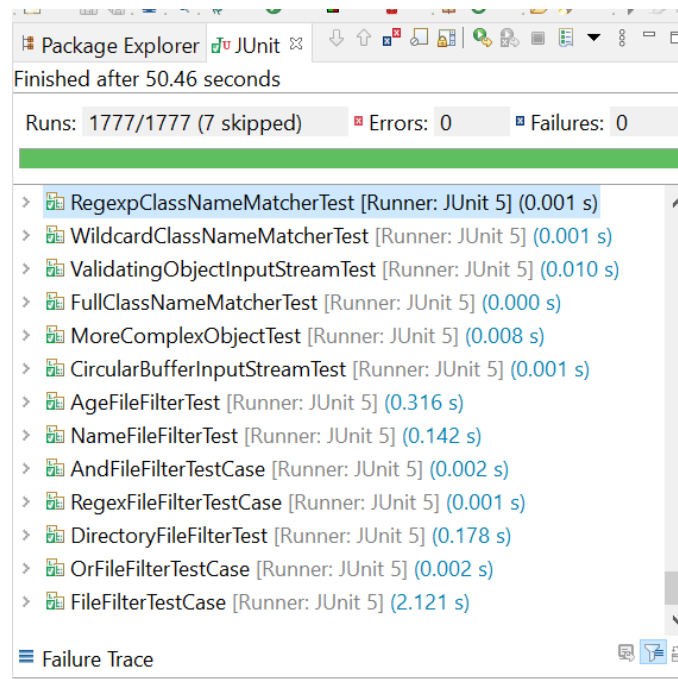

Figure 6: Results of running the whole test suite.

0.5   Covert one of your previous Java projects into Maven project

The last task of lab 0 is to covert your own Java projects into Maven project. You can follow the steps above for this task.