EECS2030 (Sec. E) Summer 2025
Lab 0 Part 1
Review on Object-Oriented Programming in Java

Chen-Wei Wang

Release Date: Tuesday, May 6
Due Date: 11:59 PM, Tuesday, May 13

Texts in blue are hyperlinks to the corresponding documents/recordings.

Follow the instructions to submit (via the web submit link) the required files for grading. **Emailing your solutions to the instructor or TAs will not be accepted**.

Policies

- **Your (submitted or un-submitted) solution to this lab exercise (which is not meant for 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. The department reserves the right to take necessary actions upon found violations of this policy.**
- Refer to the course syllabus (under the **Course Policies** section) for how labs are supposed to be attempted.
- You are responsible for **completing your submission by the set deadline** (in order to receive the credit).
- Back up your work **periodically**, so as to minimize the damage should any sort of computer failures occur. Follow this tutorial series on setting up a **private** Github repository for your Java projects.
- The deadline is **strict** with no excuses: late submissions will **not** be accepted.

Contents

Learning Outcomes

By completing the assigned exercises of this lab, you are expected to be able to:

1. Use simple Linux commands on a terminal.
2. Exercise a simple workflow of Github.
3. Understand the Observe-Model-Execute process of object orientation.
4. In the Eclipse IDE (Integrated Development Environment):
   - Create a new Java project.
   - Create a new Java class serving as a console application (with the `main` method).
   - Understand the separation of concerns (using packages): `model`, `console_apps`, and `junit_tests`.
   - Write Java `model` classes composed of:
     - classes, attributes, methods (constructors, accessors, and mutators)
     - `new` keyword, `this` keyword, dot notation
   - Run a Java class with with the `main` method as a console Java application.
   - Write JUnit test cases manipulating the model classes via *assertions*.
   - Use the debugger to:
     - Examine the object *state* (i.e., attribute values)
     - Explore the program behaviour (via step over, step into, step out)
   - Export an existing project as an archive file.
5. On paper, visualize object creations and method calls.

1 Setting Working Environment: Eclipse, Remote Labs, Github

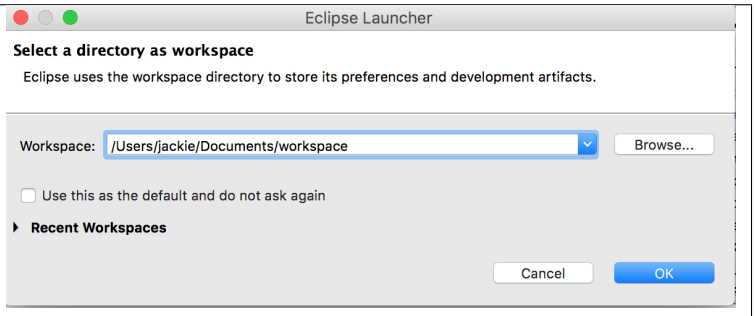
Here is a video walking you through Tasks 1.1 to 1.4:

https://www.youtube.com/watch?v=sYAcSyeoio8&list=PL5dxAmCmjv_6wy2m0yq2wObIWPz4tAxW6&index=1

1.1 Task 1.1: Install and Launch Eclipse on Your Own Computer

There is one item to install on your laptops in order to construct Java programs:

1. Eclipse IDE 2025-03 R (use the Eclipse Installer 2025-03 R) [click on the blue link]
2. When prompted, choose to install "Eclipse IDE for Java Developers".

Step 1: Launch Eclipse by clicking on its icon.	**Step 2**: You should be able to see an initial prompt like:
	

- If you are able to see the above Eclipse initial prompt, then you are ready! See the next section for tutorial videos to get started with programming in Java.

1.2 Task 1.2: Create an EECS Account

Lab submissions will require you to login using a valid EECS account. If you have not obtained one, activate it here: https://webapp.eecs.yorku.ca/activ8.

1.3 Task 1.3: Use the Remote Lab as a Backup

- You may use the Eclipse installed on your own machine to complete all lab exercises.
- However:
  - If you choose to complete a lab on your own machine, before submitting it, you are responsible for ensuring that it runs also on the version of Eclipse installed on the remote lab (by exporting it from your machine and importing it to your remote lab account).
  - The Eclipse installation on your machine may cease to work, e.g., when your lab is due soon or when your programming test is about to expire, in which case you are expected to:
    - Have a backup of your work (e.g., Github)
    - Switch to the work environment over the remote lab: https://remotelab.eecs.yorku.ca/

1.4 (Optional) Task 1.4: Create a Github Account to Manage Software Projects

- Though it is not required, it is highly recommended that you adapt to to the practice of managing your software projects using Github.
- If you already have a Github account, make sure that you are able to create **private** repositories. Otherwise, you are still required to create an **Educational** account for this course.
- Follow this tutorial series to learn about applying and using an Educational Github account (and creating **private** repositories):

  Note. Github has evolved since these tutorial videos were recorded. You might find some discrepancies, but it should still be fairly straightforward to complete the steps.

  https://www.youtube.com/playlist?list=PL5dxAmCmjv_58KxTSd1CRbpinmSF8EPJx

Notes:

- **Skip** Video 05, which is not applicable for this course.
- Video 06 is for you to synchronize between your Github repositories and your own computer. It is assumed that you already installed the *Github desktop* program on your own computer.

2 Task 2: Complete Tutorial Videos

- Here is a tutorial series for you to review object-oriented programming (OOP) in Java, and to obtain hands-on experience on Eclipse IDE (required for this course):

  https://www.eecs.yorku.ca/~jackie/teaching/tutorials/index.html#refurbished_store

- As you study through the example Java classes in the tutorial videos, you are required to **type them out**:
  - **Spellings of names of the packages, files, classes, and methods must be exact (and case-sensitive).**
  - Feel free to add comments or new Java classes/methods to experiment.
- **To submit for Part 1 of your Lab0, complete the study of 14 tutorial videos (Parts 0 – 13). You are expected to re-produce all Java classes demonstrated in the tutorials.**
- **No additional exercises are required for Lab0P1 submission.**
- You can find the iPad notes illustrated in the tutorial videos here:

  https://www.eecs.yorku.ca/~jackie/teaching/tutorials/notes/Building%20an%20Apple%20Refurbished%20Store%20App%20in%20Java.pdf

3 Task 3: Study Materials

- Here is a list of **required** materials to study alongside with the review tutorials (Section 2):

  1. Slides on Classes and Objects:

     https://www.eecs.yorku.ca/~jackie/teaching/lectures/2025/S/EECS2030/slides/01-Classes-and-Objects.pdf

  2. Written Notes:
     - Inferring Classes/Methods from JUnit Tests:

       https://www.eecs.yorku.ca/~jackie/teaching/lectures/2025/S/EECS2030/notes/EECS2030_S25_Inferring_Classes_from_JUnit.pdf
     - Declaring and Manipulating Reference-Typed, Multi-Valued Attributes:

       https://www.eecs.yorku.ca/~jackie/teaching/lectures/2025/S/EECS2030/notes/EECS2030_S25_Tracing_PointCollectorTester.pdf

     Note. Your coming programming and written tests will be based on the above slides and written notes.

- Here is a list of **optional** materials to study:

  1. Tutorial Videos on Introducing Procedural and Object-Oriented Programming in Java:

     https://www.eecs.yorku.ca/~jackie/teaching/tutorials/index.html#java_from_scratch_w21

     Specially:
     - Week 1: **Eclipse** work environment
     - Week 2c, 2d, 2e: **Debugger** in Eclipse
     - Weeks 2, 3: Programming/Debugging **Conditionals**
     - Weeks 4, 5: Programming/Debugging **Arrays** and **Loops**
     - Weeks 6, 7, 8: **Classes** and **Objects**
  2. Lecture materials (recordings, slides, notes, codes) for EECS1022-W21:

     https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS1022_W21

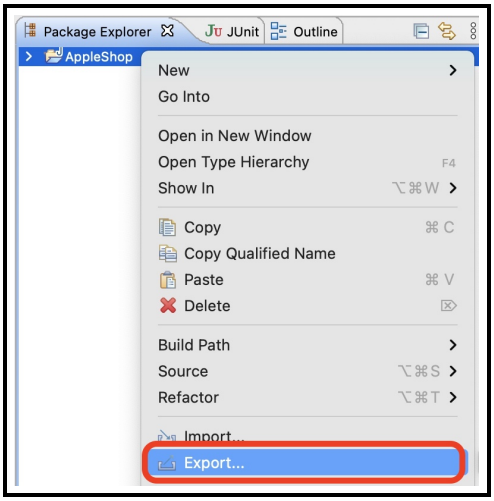
     You may refer to the relevant parts for which you find the need to review.
  3. Basics of procedural (non-OO) programming in Java:
     - Elementary Programming: https://www.eecs.yorku.ca/~jackie/teaching/lectures/2025/S/EECS2030/slides/00.01-Elementary-Programming.pdf
     - Selections: https://www.eecs.yorku.ca/~jackie/teaching/lectures/2025/S/EECS2030/slides/00.02-Selections.pdf
     - Loops: https://www.eecs.yorku.ca/~jackie/teaching/lectures/2025/S/EECS2030/slides/00.03-Loops.pdf

4 Exporting the Completed Project

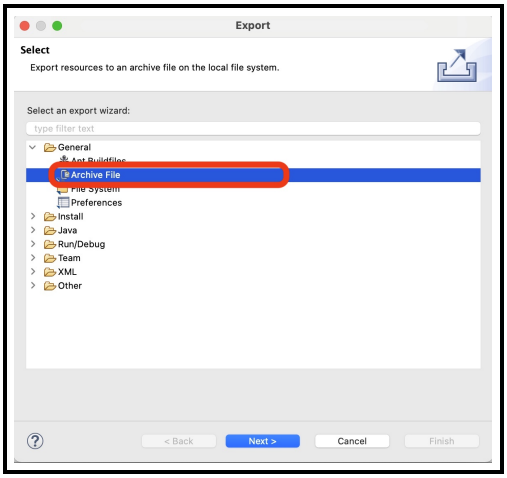
You are required to submit a Java project archive file (.zip) consisting all subfolders.

In Eclipse:

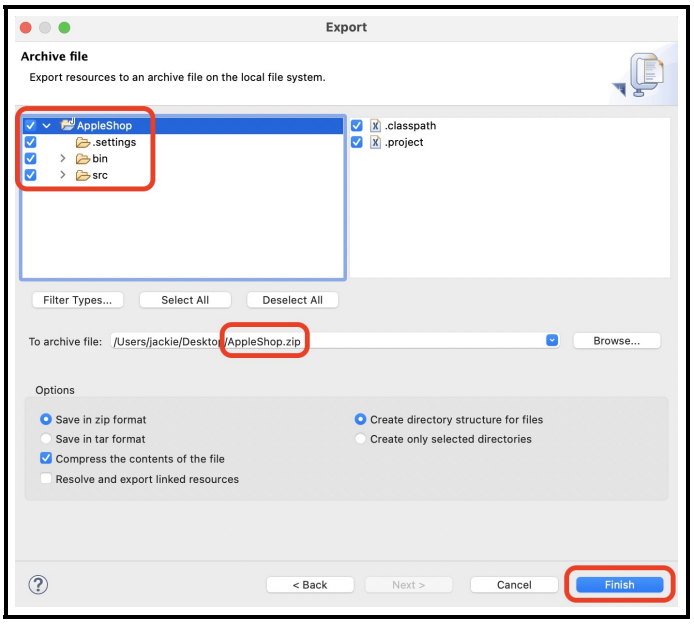
1. Right click on project **AppleShop**.
Then click **Export**



2. Under **General**, choose **Archive File**.



3. Check the top-level **AppleShop** .
Make sure that all subfolders are checked: **.settings**, **bin**, and **src**.
Under **To archive file:** browse to, e.g., desktop, and save it as **AppleShop.zip** (**case-sensitive**)
Then **Finish**.



Note. In case you have concerns about exporting and submitting the **.setting** subfolder: it will be kept confidential and access-protected on the EECS department server.

5 Submission

1. Verify that the project folder has the following structure:

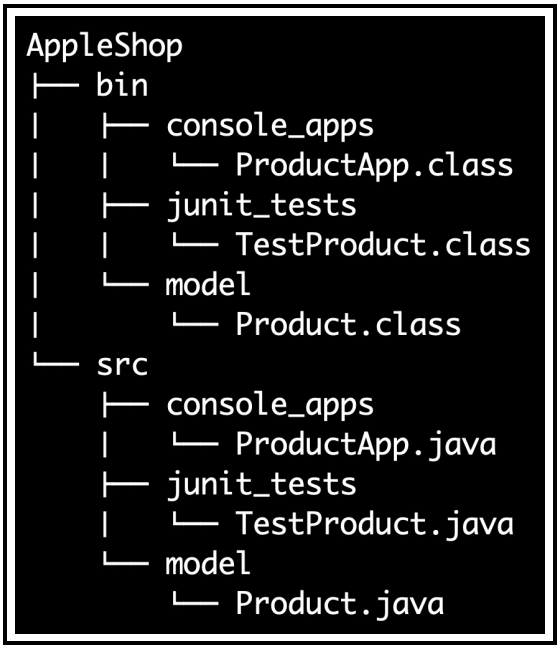


Figure 1: Lab0P1 Expected Project Structure

Names of the packages, files, classes, and methods must be exact (and case-sensitive).

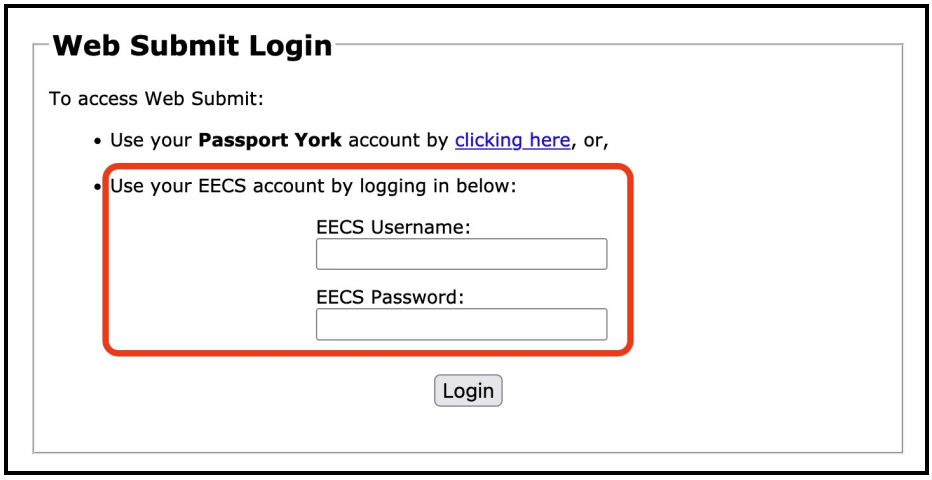
e.g., If you have a class `product.java` rather than `Product.java`, it's considered as wrong.

2. From Section 4, you have already exported your Java project on Eclipse as an archive file: `AppleShop.zip`.

   Click on the following link (for which you will be prompted to enter your EECS account login credentials):

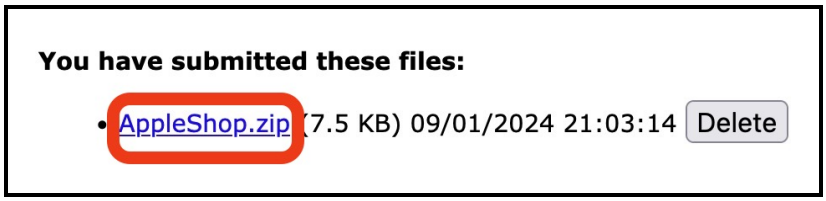
   https://webapp.eecs.yorku.ca/submit/?acadyear=2024-25&term=S&course=2030E&assignment=Lab0P1

   - You **must** login into the web submit page using your EECS login credentials (otherwise, your submitted folder on the EECS server may not be identified properly):



Note. If you are prompted for your PPY login instead, then it might be due to an earlier login session. In this case, login first with your PPY account credentials, then **log out**. Then, clicking on the above submission link should lead you to the login page for EECS account credentials.

- Ensure that the correct academic year, term, course, and assignment are chosen. Then, browse to the archive file `AppleShop.zip` and click on `Submit Files`. Be sure that the correction information is chosen:
  - **Academic Year**: `2024--25`
  - **Term**: `S`
  - **Course**: `2030E`
  - **Assignment**: `Lab0P1`
- You may upload as many draft versions as you like before the deadline.
- It is your **sole responsibility** to **download and ensure** that:
  - The submitted zip file is your best attempt (e.g,. non-empty, not the starter code).
  - The unzipped project conforms to the structure as specified in Figure 1.



6 Appendix: Basic Commands on a Terminal

- `cd`

  Change to a directory

  e.g., `cd ~`

  e.g., `cd ~/Desktop`

  e.g., `cd ~/Desktop/EECS2030-S25-workspace`

- `pwd`

  Return the path of the current directory.

- `ls`

  List the contents of the current directory.

7 Appendix: A Simple Github Workflow

- Create a **private** repository (e.g., `EECS2030-S25-workspace`) dedicated for all labs of this course.
- Clone a copy of some repository:

  ```
  git clone ...
  ```

  where ... is the URL of the repository that can be copied from a web browser.

- Make some changes to the clone copy (e.g., a new Java project, a new Java class, a change to an existing Java class).
- Add all changes to the local change list:

  ```
  git add *
  ```

- Commit the change list **locally**:

  ```
  git commit -m "..."
  ```

  where ... is some meaningful and informative log message.

- Push the local committed change to the online repository:

  ```
  git push
  ```

  Then enter your git account username and password.