# EECS2030 (Sec. E & F) Fall 2024 Lab 0 Part 1 Review on Object-Oriented Programming in Java

### Chen-Wei Wang

<u>Release</u> Date: Tuesday, September 3 <u>Due</u> Date: 12 noon, Friday, September 13

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

Your lab assignment is **not** graded during the weekly scheduled lab sessions.

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 <u>completing your submission by the set deadline</u> (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  $\underline{not}$  be accepted.

## Contents

1 Setting Working Environment: Eclipse, Remote Labs, Github			
	1.1 Task 1.1: Install and Launch Eclipse on Your Own Computer	3	
	1.2 Task 1.2: Create an EECS Account	3	
	1.3 Task 1.3: Use the Remote Lab as a Backup	3	
	1.4 (Optional) Task 1.4: Create a Github Account to Manage Software Projects	4	
<b>2</b>	Task 2: Complete Tutorial Videos	5	
3	Task 3: Study Materials	6	
4	Exporting the Completed Project	7	
5	Submission	8	
6	Appendix: Basic Commands on a Terminal	10	
7	Appendix: A Simple Github Workflow	10	

## 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 <u>debugger</u> 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, <u>visualize</u> 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 2024-06 R (use the Eclipse Installer 2024-06 R)

[click on the blue link]

2. When prompted, choose to install "Eclipse IDE for Java Developers".

<b>Step 1</b> : Launch Eclipse by clicking on its icon.	<b>Step 2</b> : You should be able to see an initial prompt like:
	Eclipse Launcher
	Select a directory as workspace
	Eclipse uses the workspace directory to store its preferences and development artifacts.
	Workspace:       /Users/jackie/Documents/workspace <ul> <li>Browse</li> <li>Use this as the default and do not ask again</li> <li>Recent Workspaces</li> </ul>
	Cancel OK

• 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):

<u>Note</u>. 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 casesensitive), as we will run some automatic checks to grade the submitted files.
  - Feel free to add comments or new Java classes/methods to experiment.
- To submit for <u>Part 1</u> of your Lab0, complete the study of 14 tutorial videos (Parts 0 13). You are expected to re-produce <u>all</u> 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/2024/F/EECS2030/slides/
01-Classes-and-Objects.pdf
```

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

https://www.eecs.yorku.ca/~jackie/teaching/lectures/2024/F/EECS2030/notes/ EECS2030\_F24\_Inferring\_Classes\_from\_JUnit.pdf

- Declaring and Manipulating Reference-Typed, Multi-Valued Attributes:

https://www.eecs.yorku.ca/~jackie/teaching/lectures/2024/F/EECS2030/notes/ EECS2030\_F24\_Tracing\_PointCollectorTester.pdf

Note. Your coming <u>programming</u> and <u>written</u> 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/2024/F/ EECS2030/slides/00.01-Elementary-Programming.pdf
  - Selections: https://www.eecs.yorku.ca/~jackie/teaching/lectures/2024/F/EECS2030/slides/ 00.02-Selections.pdf
  - Loops: https://www.eecs.yorku.ca/~jackie/teaching/lectures/2024/F/EECS2030/slides/00.
     03-Loops.pdf

#### Exporting the Completed Project 4

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

In Eclipse:

-

<ol> <li>Right click on project AppleShop. Then click Export</li> </ol>	2. Under General, choose Archive File.
Itent check DAport         Image: Explorer IX         JU JUnit         Explosion         Image: Copy Copy         Image: Copy	Export         Select         Export resources to an archive file on the local file system.         Select an export wizard:         * General         * Genera         * Genera     <

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.

Ex Ex	port	
Archive file Export resources to an archive file on the local file system.		
<ul> <li>✓ ✓ AppleShop</li> <li>✓ → Settings</li> <li>✓ → bin</li> <li>✓ → bin</li> <li>✓ → bin</li> <li>✓ → bin</li> </ul>	<ul> <li>☑ classpath</li> <li>☑ .project</li> </ul>	
Filter Types Select All Deselect All To archive file: /Users/jackie/Deskto AppleShop.zip Options		Browse
<ul> <li>Save in zip format</li> <li>Save in tar format</li> <li>Compress the contents of the file</li> <li>Resolve and export linked resources</li> </ul>	Create directory structure for files     Create only selected directories	
< Back	Next > Cancel	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), as we will run some automatic checks to grade the submitted files.

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

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 <u>EECS account</u> login credentials):

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

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

Web Submit Login				
Use your <b>Passport York</b> account by <u>clicking here</u> , or,				
Use your EECS account by logging in below:				
	EECS Username:			
	EECS Password:			
Login				

**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 <u>log out</u>. 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.

Acaden	Academic Year: 2024-25 V			
	Term: F 🗸			
	Course:	2030EF ~		
Assi	Assignment: Lab0P1 v			
Submit Status: Submission				
	Enabled			
Fe	Feedback: None			
Please specify files to submit: (You can submit multiple files at once!)				
Browse	se AppleShop.zip			
Browse	se No files selected.			
Browse	No files s	elected.		
Browse	No files s	elected.		
Browse	Browse No files selected.			
Browse	Browse No files selected.			
Browse	Browse No files selected.			
Browse	Browse No files selected.			
Browse	Browse No files selected.			
Browse	Browse No files selected.			
	Submit Files Logout			

- You may upload <u>as many draft versions as you like</u> before the deadline only the <u>latest</u> submitted version of your work <u>before the deadline</u> will be graded.
- It is your **sole responsibility** to **download and ensure** that:
  - The submitted zip file is the one you intend to be graded (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-F24-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-F24-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.