

# EECS1022 (M,N,O) Winter 2021

## Lab 0

### Managing Software using Github and Java Programming in Eclipse

Chen-Wei Wang

**Release Date: Monday, January 11**  
**Due Date: 23:59 EST, Friday, January 15**

[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 eClass) the required files for grading. Emailing your solutions to the instructor or TAs will not be acceptable.

## Policies

- **Your (submitted or un-submitted) solution to this lab exercise (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. The department reserves the right to take necessary actions upon found violations of this policy.**
- When you submit your lab, 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 Java code during **any** stages of your development.
- When assessing your submission, the instructor and TA may examine your code, and suspicious submissions will be reported to the department if necessary. **We do not tolerate academic dishonesty**, so please obey this policy strictly.
- You are entirely responsible for making your submission to the TA in time. 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. Refer to the course syllabus for the policy on a late submission.

## Contents

1	Task 1: Creating a Github Account to Manage Software Projects	3
2	Task 2: Installing and Launching Eclipse on Your Own Computer	3
3	Task 3: Tutorial Videos to Complete	4
4	Submission	4
5	Appendix: Basic Commands on a Terminal	5
6	Appendix: A Simple Github Workflow	5

## 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. 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).
  - Create a new Java class serving as a holder of utility methods.
  - Write Java programs composed of:
    - Numerical Literals and operators
    - String Literals and operators
    - Variables
    - Console outputs (via `System.out.println`)
    - Keyboard inputs (via `nextInt()`, `nextDouble()`, and `nextLine()` of `Scanner`)
  - Run a Java class with with the `main` method as a console Java application.
  - Implement utility methods (called by the console applications).
  - Write a JUnit test (calling the utility methods)
  - Export an existing project as an archive file.
4. Understand the separation of concerns (using packages): `model`, `console_apps`, and `junit_tests`.

## 1 Task 1: Creating a Github Account to Manage Software Projects

- Though it is not required, it is highly recommended that you adapt 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.

**Requirement:** You **must** create an **Educational** Github account, e.g., with a user name structured as follows (separated by single hyphens):

lastName-firstName

For example: **smith-john** would be the Github account name for a student whose last name is Smith and first name is John.

- Follow this tutorial series to learn about applying and using an Educational Github account (and creating **private** repositories):

[https://www.youtube.com/playlist?list=PL5dxAmCmjv\\_58KxTSd1CRbpinmSF8EPJx](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: Installing and Launching Eclipse on Your Own Computer

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

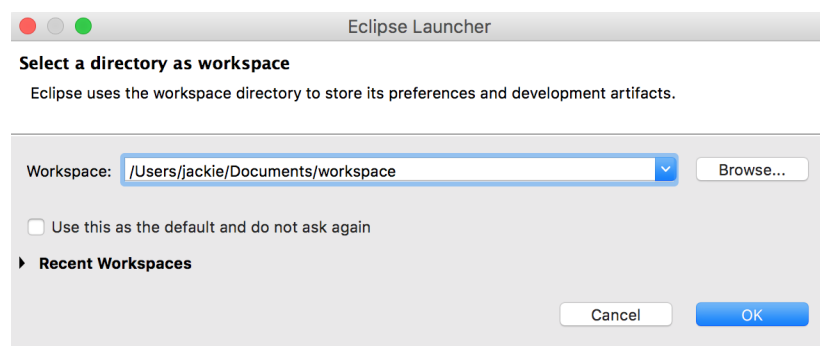
[Eclipse IDE 2020-12](#) (using the Eclipse Installer 202012 R)

[click on the blue link]

To launch Eclipse, simply click on its icon:



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.

### 3 Task 3: Tutorial Videos to Complete

- Here is a tutorial series for you to obtain hands-on experience on the basics of Java programming:

[https://www.youtube.com/playlist?list=PL5dxAmCmjv\\_6wy2m0yq2w0bIWPz4tAxW6](https://www.youtube.com/playlist?list=PL5dxAmCmjv_6wy2m0yq2w0bIWPz4tAxW6)

The above tutorial series will be expanded with videos on a weekly basis to supplement your lectures.

- As you study through the example Java classes in the tutorial videos, you are required to **type them out** (but absolutely feel free to add new Java classes to experiment) on Eclipse.
- **To submit for Lab0, complete the study of Week 01 videos (Parts A – F). You are expected to re-produce all Java classes demonstrated in the tutorials.**

**Spellings of 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.**

- **No additional exercises are required for Lab0 submission.**
- You can find the iPad notes illustrated in the tutorial videos here:

<https://www.eecs.yorku.ca/~jackie/teaching/tutorials/notes/EECS1022%20Tutorial%20on%20Java.pdf>

### 4 Submission

1. Part F of Week 01's tutorial videos asks you to **export** the Java project as an archive file:

EECS1022.W21.Week\_01.zip

Before you submit, verify that its unzipped version has the following structure (the **bin** directory may be ignored; the **src** folder structure must be **exact**):

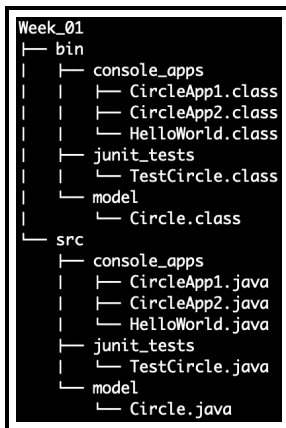


Figure 1: Lab0 Expected Project Structure

**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 **ConsolApp\_2** rather than **ConsoleApp2**, it's considered as wrong.

2. Go to the eClass site for Sections M,N,O: <https://eclass.yorku.ca/eclass/course/view.php?id=6214>
3. Under the **Lab Submissions** section, click on **Lab0** to submit the Java archive file:

EECS1022.W21.Week\_01.zip:

- You may **upload** as many draft versions as you like before the deadline.
- You must explicitly **submit** the draft version for grading before the deadline.
- **Once you click on the submit button, you can no longer upload another draft version.**

## 5 Appendix: Basic Commands on a Terminal

- **cd**

Change to a directory

e.g., `cd ~`

e.g., `cd ~/Desktop`

e.g., `cd ~/Desktop/EECS1022-W21-workspace`

- **pwd**

Return the path of the current directory.

- **ls**

List the contents of the current directory.

## 6 Appendix: A Simple Github Workflow

- Create a **private** repository (e.g., `EECS1022-W21-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.