EECS1022 (M,N,O) Winter 2021:
Programming for Mobile Computing
Programming Test 1

Requirements of this Lab Test

- This programming test is **strictly** individual: plagiarism check will be performed on all submissions, and suspicious submissions will be reported to Lassonde for **a breach of academic honesty**.
- You are given **90 minutes** to complete the submission. The time limit is **strict** so you are solely responsible for leaving enough time (e.g., 10 minutes) to export the completed Java project and upload/submit the archive (.zip) file to eClass.
- If your submitted Java class does not compile, you receive a **penalty** as specified in your test guide.

  It is therefore absolutely critical for you to always make sure that there's not any red-underlines in your Java classes.
- Your submission will only be graded by:
  - JUnit tests given to you in `TestUtilities.java`
  - additional JUnit tests on input values not covered in `TestUtilities`

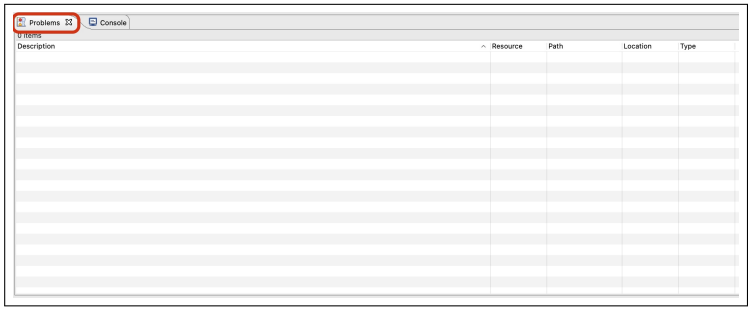
Reminders of Some Java Syntax

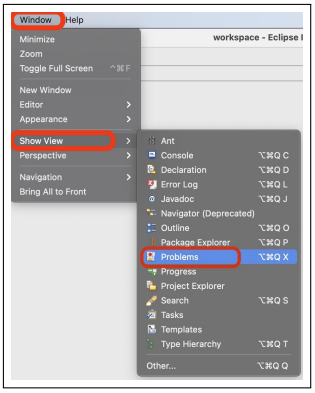
- To compare primitive values (e.g., `int`, `double`, `char`, *etc.*), use the == relational operator.
- To compare the contents of two strings `s1` and `s2`, write `s1.equals(s2)`.

1 Getting Ready for Submission

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

1. Before you submit, you must make sure that the `Problems` panel on your Eclipse shows **no errors** (warnings **are** acceptable). In case you do not see the `Problems` panel: click on `Window`, then `Show View`, then `Problems`.



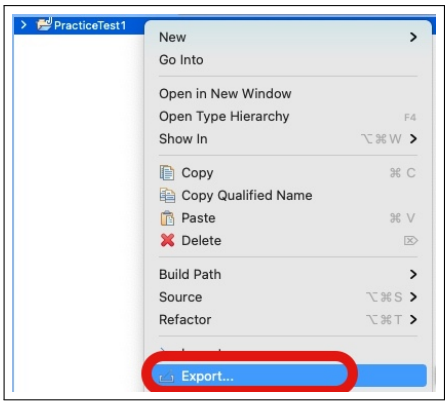


Submitting programs with errors (meaning that it cannot be run for grading) will result in possible partial, but low, marks.

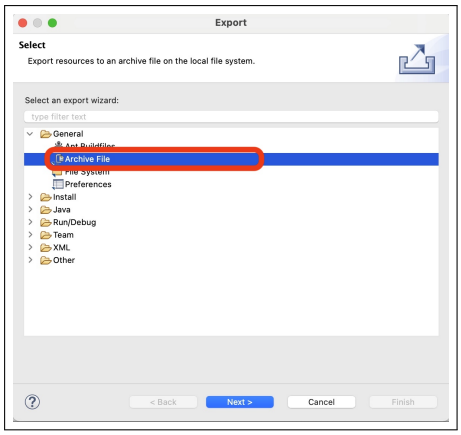
2 Submission

In Eclipse:

1. Right click on project `Test1c`. Then click `Export`



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

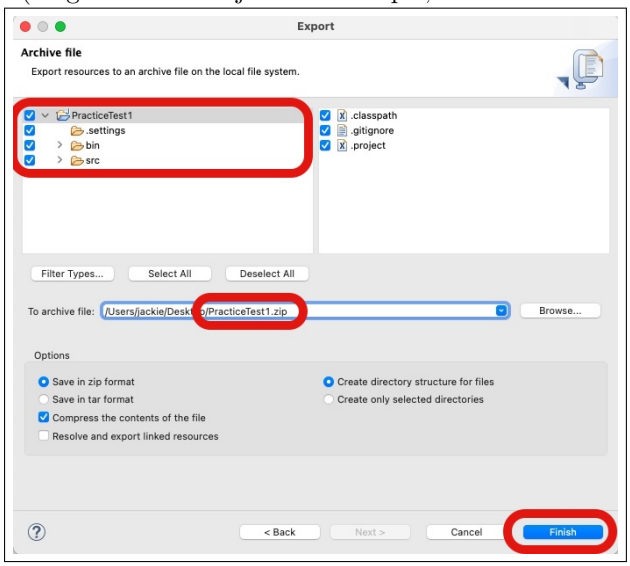


3. Check the top-level `Test1c`

Make sure that all subfolders are checked: `.settings`, `bin`, and `src`.

Under `To archive file:` browse to, e.g., desktop, and save it as `Test1c.zip` (**case-sensitive**)

Then `Finish`. (diagram below is just for example; name should be `Test1c.zip`)



2. Go to the eClass site for Sections M,N,O: `https://eclass.yorku.ca/eclass/course/view.php?id=6214`

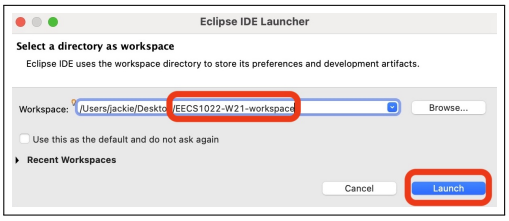
3. Attach the Java archive file: `Test1c.zip`

   - 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.**

3 Getting Started

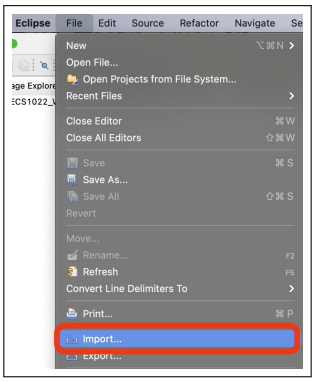
3.1 Step 1: Download and Import the Starter Project

1. Download the Eclipse Java project archive file from eClass: `Test1c.zip`
2. Launch Eclipse and browse to `EECS1022-W21-workspace` as the `Workspace` then click on `Launch`, e.g.,

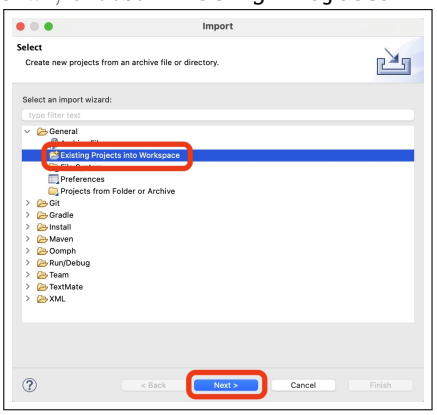


3. In Eclipse:

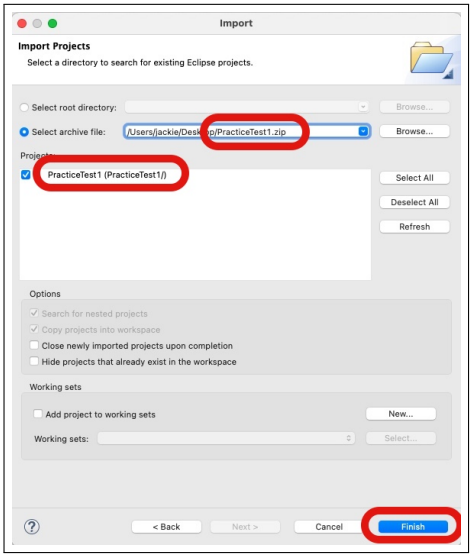
3.1 Choose `File`, then `Import`.



3.2 Under `General`, choose `Existing Projects into Workspace`.



3.3 Choose `Select archive file`. Make sure that the `Test1c` box is checked under `Projects`. Then `Finish`. (diagram below is just for example; name should be `Test1c.zip`)



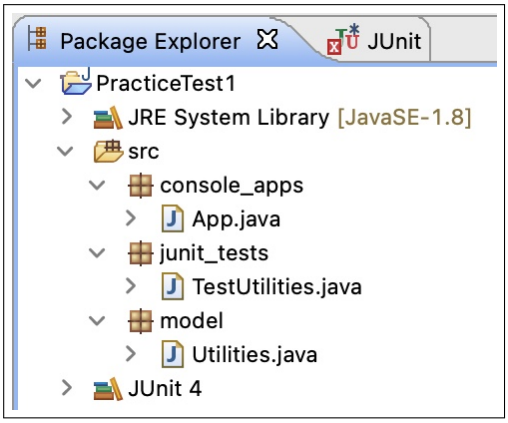
3.2 Step 2: Programming Tasks

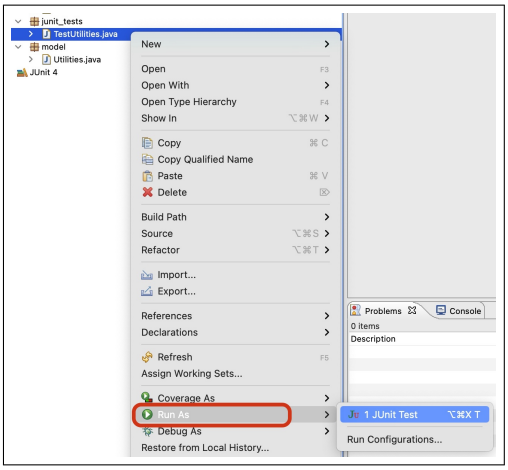
From the `Package Explorer` of Eclipse, your imported project has the following structure.

- Your submission will only be graded by:
  - JUnit tests given to you in `TestUtilities.java`
  - additional JUnit tests on input values not covered in `TestUtilities`
- You may manually test the assigned methods using the `App` console application class given to you in `console_apps`. Declaration of the `main` method and a scanner are completed for you.
- Your goal is to pass **all** JUnit tests given to you (i.e., a **green bar**). To run them, as shown in the Java tutorials on Week 1, right click on `TestUtilities.java` and run it as JUnit tests. Of course, none of the given tests would pass to begin with.

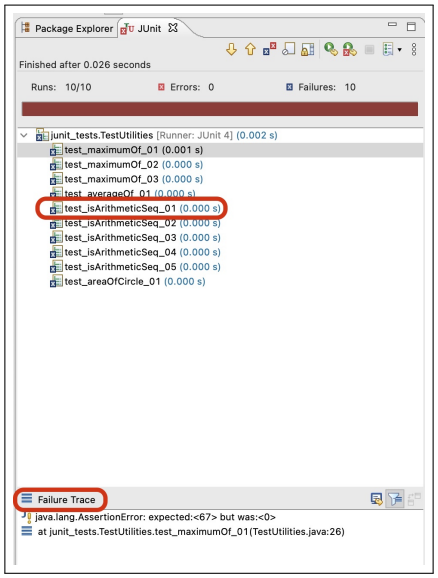
  You must not modify the JUnit test methods given to you.

  However, you are allowed to add new JUnit test methods to test your code.





How to Deal with a Failed JUnit Test? From the JUnit panel from Eclipse, click on the failed test, then **double click** on the first line underneath `Failure Trace`, then you can see the **expected value** versus the **return value** from your utility method. For example:



4 Your Tasks

- See the comments written in the `Utiliites` class to see each of the assigned methods to complete.
- See test methods in the `TestUtilities` class for examples. You may add additional test methods if you wish.