

Victoria University of Wellington
School of Mathematics, Statistics and Computer Science

SWEN221: Software Development Lab Handout

Outline

The purpose of this lab is to refresh your Java programming skills, to get started using Eclipse and to learn how to use the online submission system. Before the end of the lab, you should submit your solution to Activity 3 (the **calculator**) via the *online submission system* which will automatically mark it. You may submit as many times as you like in order to improve your mark and the final deadline will be Friday @ 23:59.

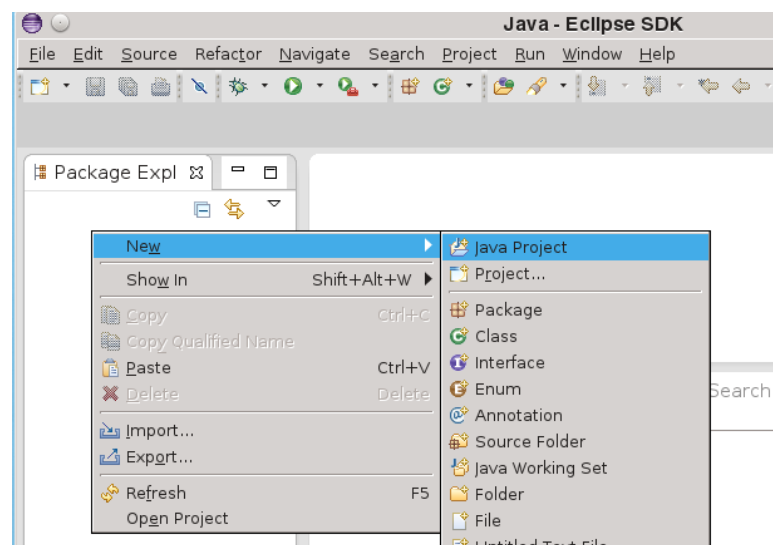
Activity 1: Diagnostic Test (approx 50mins)

(See separate test sheet)

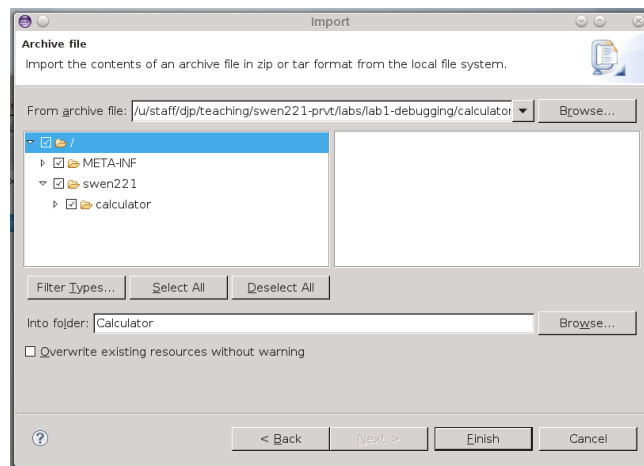
Activity 2: Getting Started with Eclipse (approx 15mins)

To begin, download the `calculator.jar` program from the lecture schedule on the SWEN221 course website. Now, perform the following steps:

1. Start Eclipse and create a project called “Calculator”:

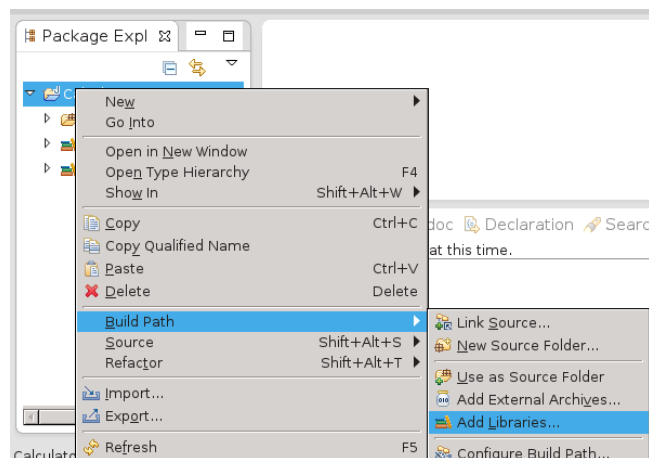


2. Import the `calculator.jar` file into your Calculator project. Do this by selecting “**File**→**Import**→**General**→**Archive File**” from the menu:



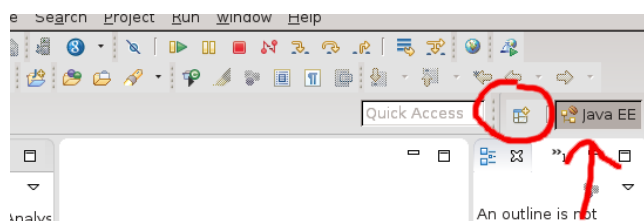
Select the jar file in the “Import” dialog, and then set the “**Into folder**” field to be “**Calculator/src**” and click “**Finished**”.

3. Add JUnit library to your project. Do this by right-clicking on the project and selecting “**Build Path**→**Add Libraries**→ **JUnit**→**JUnit 5/Jupiter**” from the menu:

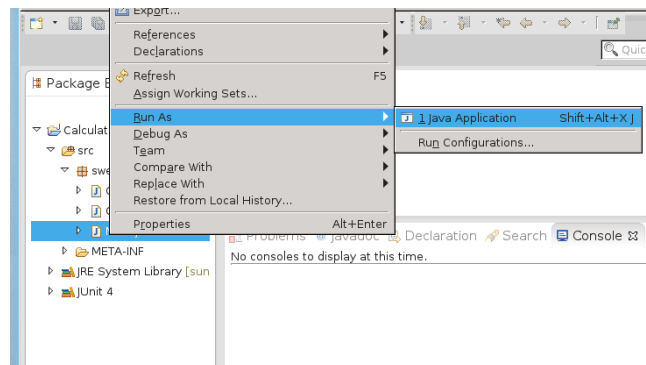


Your project should now compile without errors.

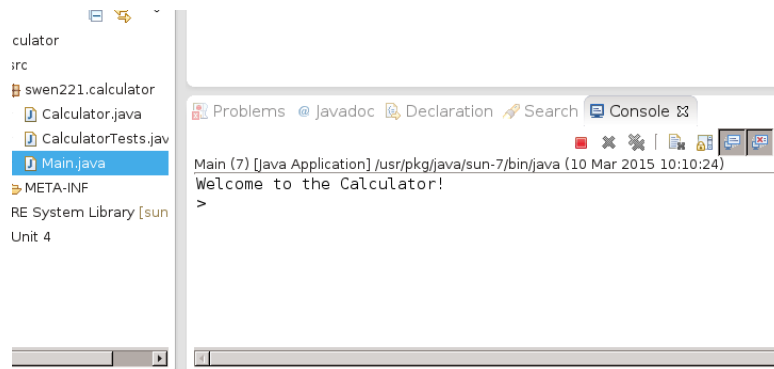
NOTE: if you cannot find the “**Build Path**” menu, then you should check that you are in the “Java” perspective, rather than the “Java EE” perspective:



4. Run the calculator program from Eclipse by creating a “Run Configuration”. One way to do this is by right-clicking on **Main** and selecting “Run As→Java Application”:



You should see the calculator running in the console window:



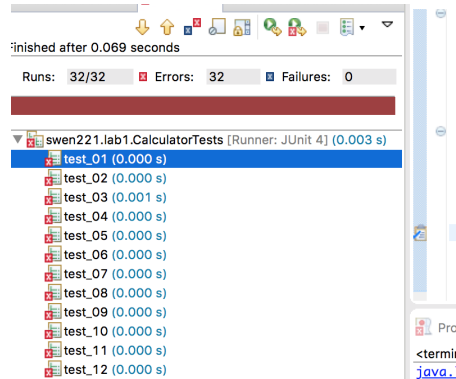
5. You can now perform simple calculations, like with a normal calculator. For example:

```
Welcome to the Calculator!
> 1.0+1.23
= 2.23
> 1+2.0
Found '+', expected '.': 1+2.0
      ^
java.lang.RuntimeException: Parse Error
at swen221.lab1.Calculator.error(Calculator.java:144)
at swen221.lab1.Calculator.match(Calculator.java:115)
...
```

Spend some time using the calculator to get the hang of it; you might notice that there are several bugs in the system (as illustrated above).

Activity 3: Debugging the Calculator (approx 45min)

Your next task is to debug the program. To get started, run the JUnit tests provided (by right-clicking on `CalculatorTests`, and selecting “**Run As→JUnit Test**”):



Several of the tests will fail and you’ll need to fix the bugs to make them pass correctly.

HINT: Simply “eyeballing the code” (i.e. staring at it) will probably not help you find the bug! You must insert `println` statements to see what is actually going on, and to narrow down the problem.

Submission

Your lab solution should be submitted electronically via the *online submission system*, linked from the course homepage. The required files are:

`swen221/lab1/Calculator.java`

You must ensure your submission meets the following requirements (which are needed for the automatic marking script):

1. **Your submission is packaged into a jar file, including the source code.** *Note, the jar file does not need to be executable.* See the following Eclipse tutorials for more on this:
<http://ecs.victoria.ac.nz/Support/TechNoteEclipseTutorials>
2. **The names of all classes, methods and packages remain unchanged.** That is, you may add new classes and/or new methods and you may modify the body of existing methods. However, you may not change the name of any existing class, method or package. *This is to ensure the automatic marking script can test your code.*
3. **All JUnit test files supplied for the assignment remain unchanged.** Specifically, you cannot alter the way in which your code is tested as the marking script relies on this. This does not prohibit you from adding new tests, as you can still create additional JUnit test files. *This is to ensure the automatic marking script can test your code.*
4. **You have removed any debugging code that produces output, or otherwise affects the computation.** *This ensures the output seen by the automatic marking script does not include spurious information.*

Note: Failure to meet these requirements could result in your submission being reject by the submission system and/or zero marks being awarded.