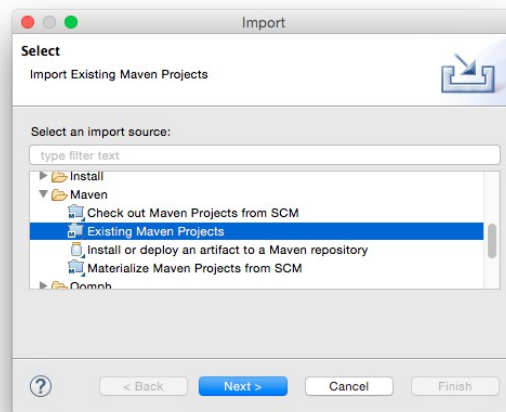


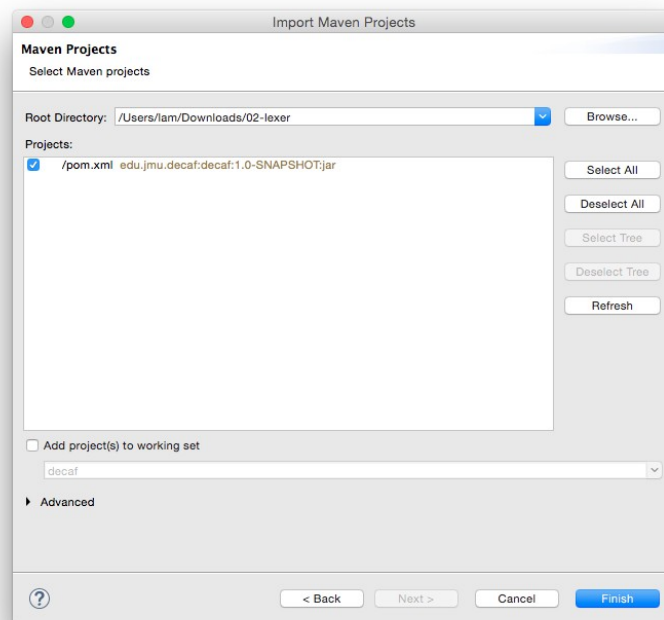
Decaf Project: Using Eclipse

IMPORTING

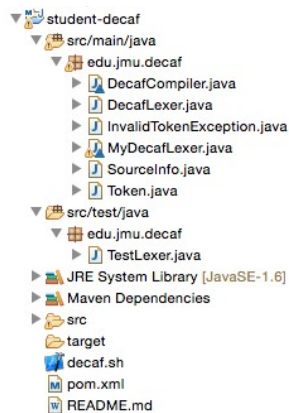
1. Download and decompress the project files.
2. In Eclipse, choose menu item File->Import.
3. Select "Existing Maven Projects" in the "Maven" folder:



4. Click Next, then click "Browse" and select the root folder of your project files. Make sure the project checkbox is checked:



5. Click "Finish". You should now have the following items in your Package Explorer:



The project is now set up. At this point, you can edit the source files in Eclipse and compile/run/test the project from the command line.

To compile:

```
mvn compile
```

To run (after compiling):

```
./decaf.sh <input-file>
```

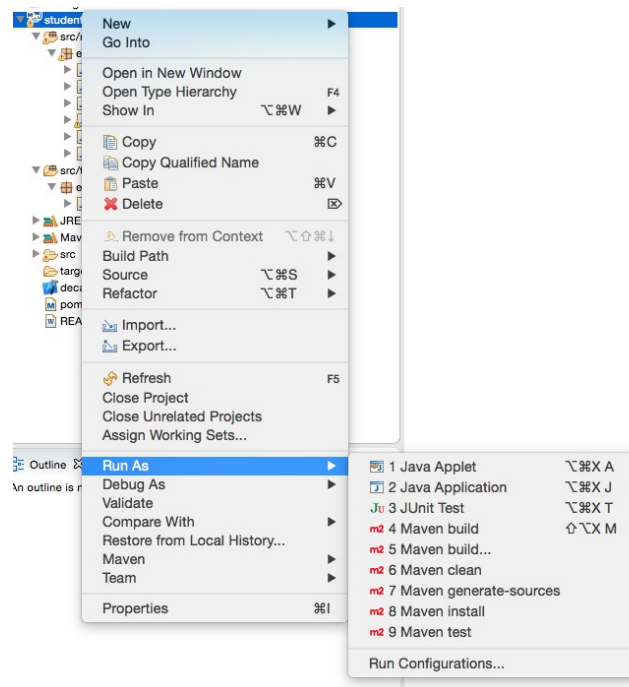
To test:

```
mvn test
```

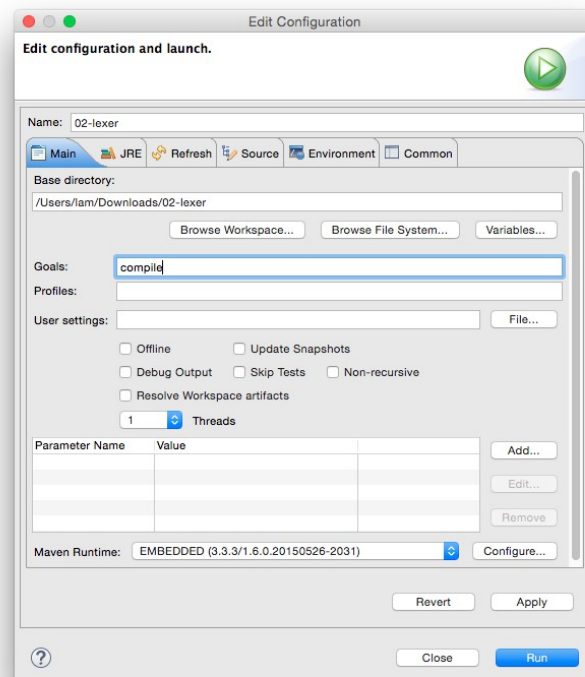
If you want, you can also set up Eclipse configurations for these tasks. The following sections describe how to do this.

COMPILING

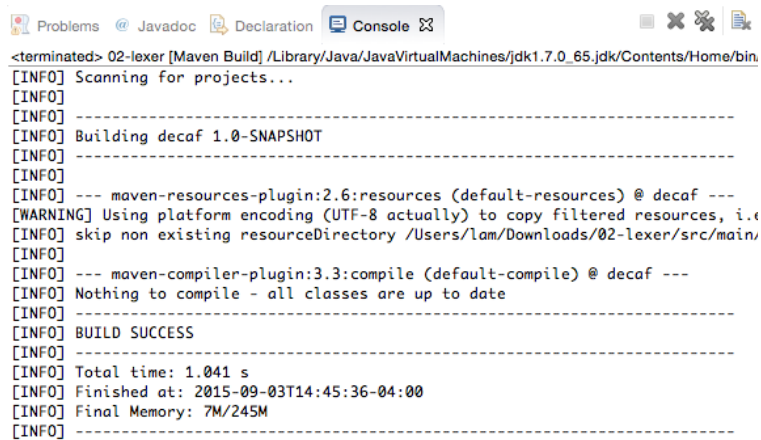
1. Right-click the project and select "Run As" → "Maven Build"



2. Enter "compile" in the "Goals" textbox. Take note of the value of the "Name" field.

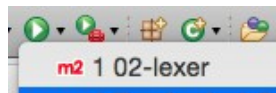


3. Click "Run". You should see the compilation process run in the Console window:



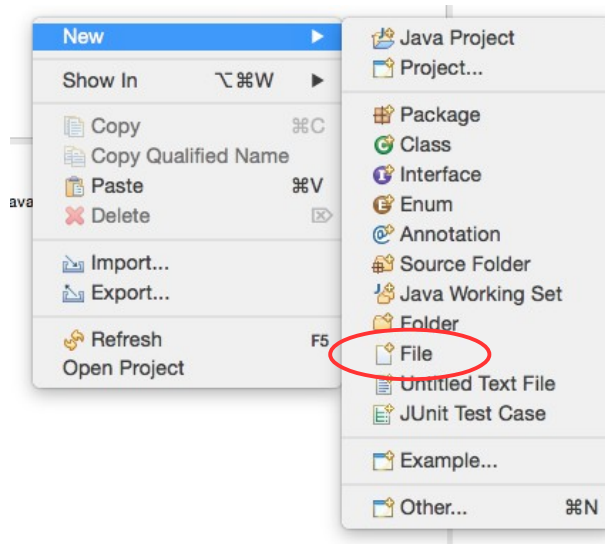
```
<terminated> 02-lexer [Maven Build] /Library/Java/JavaVirtualMachines/jdk1.7.0_65.jdk/Contents/Home/bin
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building decaf 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ decaf ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
[INFO] skip non existing resourceDirectory /Users/lam/Downloads/02-lexer/src/main
[INFO]
[INFO] --- maven-compiler-plugin:3.3:compile (default-compile) @ decaf ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 1.041 s
[INFO] Finished at: 2015-09-03T14:45:36-04:00
[INFO] Final Memory: 7M/245M
[INFO] -----
```

4. Now you can compile the project by clicking the down-arrow next to the "Run As" toolbar button and choosing the target you created in Step 2 (the label should match the "Name" field from the dialog):

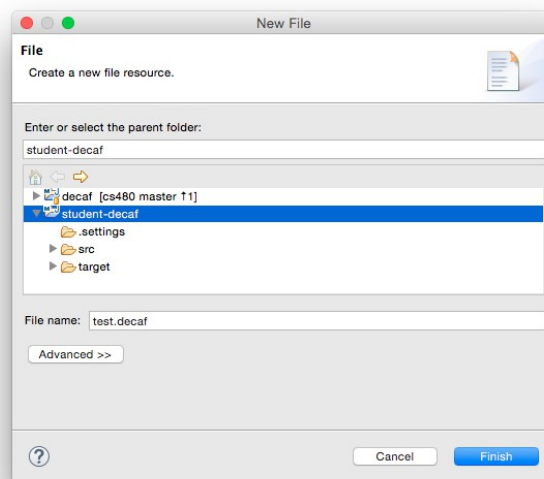


RUNNING

1. Create a test Decaf file by right-clicking in the Package Explorer and selecting New → File:



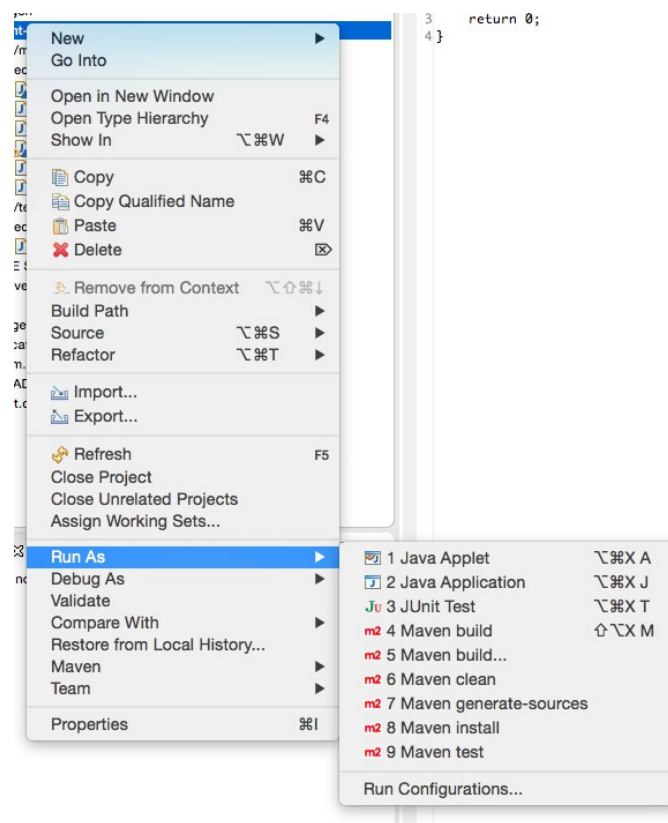
2. Choose a name (e.g., "test.decaf") and select the folder where you want the file to be located.



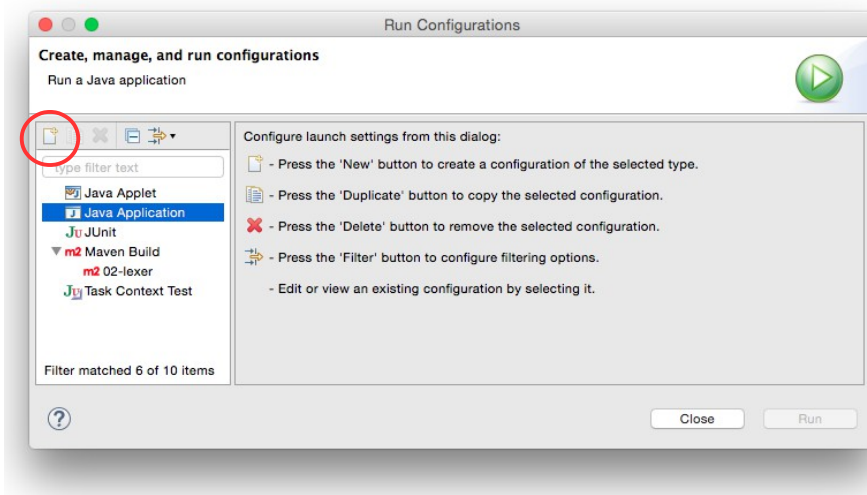
3. Write a decaf program in the new file. Here is a generic empty program:

```
def int main()
{
    return 0;
}
```

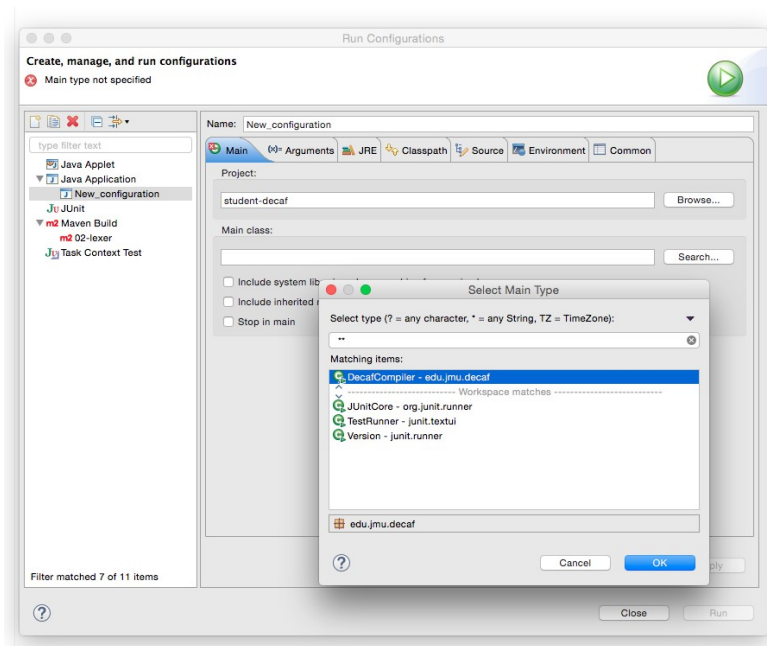
4. Right-click the project and select "Run As" → "Run Configurations":



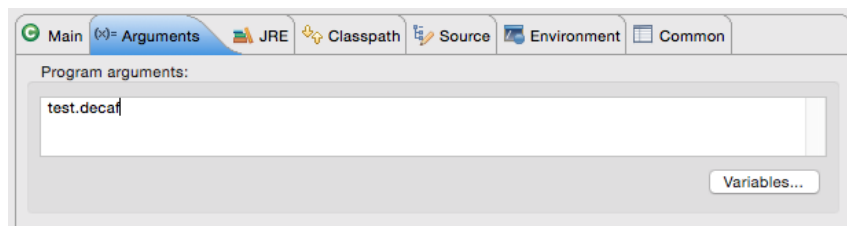
5. Make sure "Java Application" is selected and click the "New launch configuration" button:



6. Click "Search" in the "Main class" section and select the DecafCompiler class:

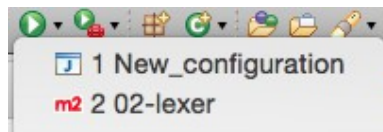


7. In the "Arguments" tab, enter the filename of the Decaf program:



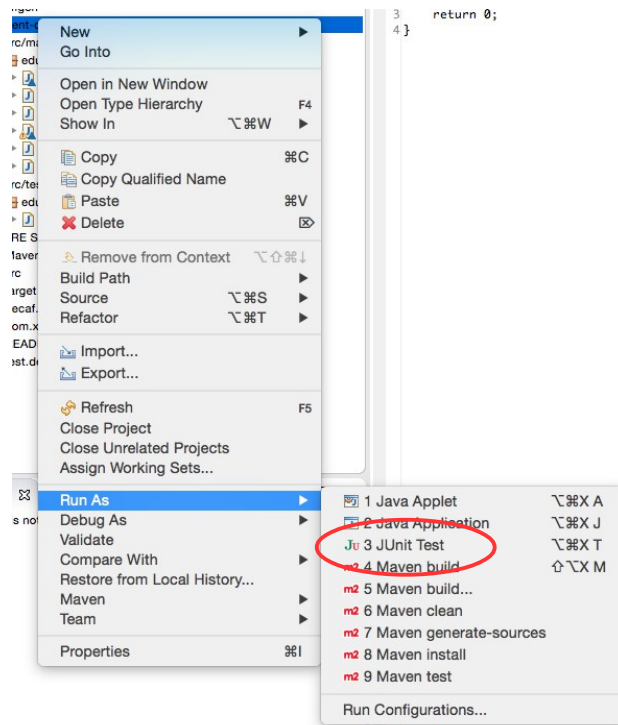
8. Enter a new name for the configuration using the "Name" field if you like, then click "Run."

9. It will save the configuration and you can run it again later by choosing it from the "Run As" menu on the toolbar:

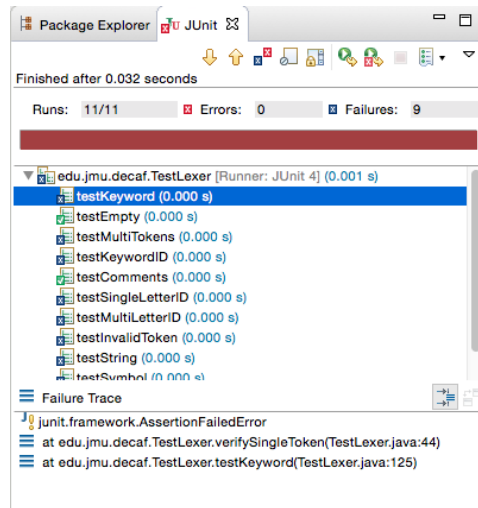


TESTING

1. Right-click the project and choose "Run As " → "JUnit Test":



2. View the results in the "JUnit" window:



3. You can run the tests again later by choosing it from the "Run As" menu on the toolbar:

