CM1209 ASSESSMENT 2: RUNNING THE UNIT TEST FILE

Unit Test File

Assessment2UnitTests.java contains the unit tests that will be used to evaluate your submission. This file is provided for your reference; it does not need to be modified and should not be uploaded to Learning Central as part of your final submission. You should check this file to determine how many marks each unit test contributes to the overall mark for this assessment.

How To Run The Unit Test File

Download the file **ast2ut.zip** from Learning Central:

CM1209 → Coursework → Unit Tests

Unzip this file into a convenient location on your computer. You will see a folder called "ast2ut" with the following file structure:

```
ast2ut
├── Assessment2UnitTests.java
├── hamcrest-core-1.3.jar
├── junit-4.12
├── TestTools.java
```

Testing You Submission

You should only try to test your code, once you have completed questions 1 **AND** 2. Trying to do so before completion of both questions could result in strange results. Once you have completed both questions, place your amended copies of **Animal.java** and **Zoo.java** into the **ast2ut** folder, which should then look like this:

```
ast2ut
├── Assessment2UnitTests.java
├── Animal.java
├── hamcrest-core-1.3.jar
├── junit-4.12
├── Zoo.java
├── TestTools.java
```

Now that we have all of the files needed, we can compile and run our unit test file to see how many tests our code has passed/failed. Open up a command window and navigate to the **ast2ut** folder, containing all your files. Then type one of the following, depending on your Operating System

On Windows

```
javac -cp .;junit-4.12.jar Assessment2UnitTests.java
java -cp .;junit-4.12.jar;hamcrest-core-1.3.jar org.junit.runner.JUnitCore Assessment2UnitTests
```

On Mac/Linux

```
javac -cp .:junit-4.12.jar Assessment2UnitTests.java
java -cp .:junit-4.12.jar:hamcrest-core-1.3.jar org.junit.runner.JUnitCore Assessment2UnitTests
```

If everything has worked correctly, you will see an output showing you how many tests your code has passed, otherwise, the failures will be listed. The following examples show the output for code that has passed ALL tests and code that has failed one test.

Output For All Tests Psssed

```
JUnit version 4.12
...............
Time: 0.013

OK (15 tests)
```

Output For One Test Failed

```
JUnit version 4.12
..............E..
Time: 0.018
There was 1 failure:
1) q2testH(Assessment2UnitTests)
java.lang.AssertionError: Zoo's fields should be hidden.
        at org.junit.Assert.fail(Assert.java:88)
        at Assessment2UnitTests.q2testH(Assessment2UnitTests.java:191)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)
        at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
        at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)
        at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
        at org.junit.internal.runners.statements.FailOnTimeout$CallableStatement.call(FailOnTimeout.java:298)
        at org.junit.internal.runners.statements.FailOnTimeout$CallableStatement.call(FailOnTimeout.java:292)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.lang.Thread.run(Thread.java:745)

FAILURES!!!
Tests run: 15,  Failures: 1
```

NOTE: Each time you make any changes to **Animal.java** and **Zoo.java**, you will need to copy the latest versions into the **ast2ut** folder and run the **java/javac** commands again, so you are running the unit tests on the latest version of your code.

REMEMBER: If you are having any difficulties running the unit tests after trying these instructions, you can also bring your source files to the following exercise laboratory sessions, where I can run them against the unit tests for you:

Monday 29th February 2016 @ 16:10
Tuesday 1st March 2016 @ 09:00