M. Roggenbach, CS-135 – Lab Class 6 – 9.3.2020

- To be solved in groups of two.
- To be ticked off in one of the labs of your house on Monday, 9.3., or Monday, 16.3.
- For being ticked off on Monday, 16.3., you need to have your solution ready at the begin of the lab class.
- You can obtain two marks by solving this sheet.
- Each completed task gives you one mark.
- All group participants need to be present to be ticked off.

This lab is about Whitebox Testing.

The purpose of this lab is to get some experience with the tool Emma.

Note that there are computer instruction at the end of this lab sheet.

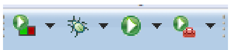
□ Task 6.1

Getting Started with Emma

1. Find out if Eclipse Plugin "Emma" has already been installed on your computer: One way of finding out is to select Help → About Eclipse Platform. From this dialog, click Plug-in Details to get a list of all installed plug-ins. If "Emma" appears in this list, continue at 3.; otherwise go to 2.
2. Install the tool (Eclipse Plugin) "Emma" from the `http://update.eclemma.org/`:
   (a) Start Eclipse.
   (b) Click `Help` → `Install New Software`
   (c) Select `Add`, give it a name such as `Emma`, type in the URL `http://update.eclemma.org/` and click OK.
   (d) Select `EclEmma`.
   (e) Click twice `Next`, accept the license, and click `Finish`.
   (f) Restart Eclipse (if needed).
3. Make a new Java Project in Eclipse.
4. Download the files `Clipper.java` and `ClipperTests.java` from

   `http://www.cs.swan.ac.uk/~csmarkus/Tools/`
5. Import these files into your Eclipse Project.
6. Activate JUnit for the project (see computer instructions) and run the JUnit test suite `ClipperTests`.

7. Run `CodeCover` by clicking the the leftmost “play” button of



   Note: You must run the normal JUnit before this step.

8. Check in the `coverage` sub-window (should it not show automatically, you can get it via `Window -> Show View -> Other -> Java -> Coverage`) that the **method** `clip` has 100% coverage.

   Note: Remember the system under test is the method `clip`. Emma shows coverage for all Java code. You need to expand the package tree and navigate down to the `clip` method in order to isolate it.

   Screen shot 1: Screen shot of eclipse showing that the **method** `clip` has 100% coverage.

9. Open the `Clipper.java` source file. Check that all statements of the `clip` method are underlined in green. This indicates that they have all been executed during the testing.

10. Remove the test cases `clip2, clip3, clip4` from `ClipperTests`. Run the tools again (JUnit and Emma). Check the coverage of `clip`: it is down to 83.3%, and the code of `Clipper.java` shows one line in red – indicating that the test suite did not cover it – and two lines in yellow – indicating that only one of two possible branches was tested.

    Screen shot 2: Screen shot of eclipse showing that the **method** `clip` has now 83.3% coverage.

Material to show when getting ticked off: Screen shot 1 and screen shot 2.

□ Task 6.2

Developing a Whitebox Test Suite with Emma

Consider the extended Mortgage-Problem, where it is computed how much a person can borrow. Inputs are *age*, *salary*, and *gender*. Output is the amount that one can borrow.

Mortgage:

Input: integer *age* in the range 18 .. 55
integer *salary* in the range 1 .. 100,000
gender ∈ {male, female}

Output: integer *salary* * *factor*, where
factor is given by the following table

Category	*factor*
Young (18–35 years)	7.5 male, 7 female
Middle (36 – 45 years)	5.5 male, 5 female
Old (46 – 55 years)	3.5 male, 3 female

1. Make a new Java Project in Eclipse.

2. Download the files `Mortgage.java` and `Main.java` from

   `http://www.cs.swan.ac.uk/~csmarkus/Tools/`

3. Import these files into your Eclipse Project and run the `main` method.

4. Develop a White Box Test suite with 100% coverage for the method `calculateMortgage`.

   Note: You might want to draw the program graph of `calculateMortgage.java` in order to systematically develop the test cases for C_p coverage.

Material to show when getting ticked off: Screen shot that you have reached full coverage and your JUnit code.

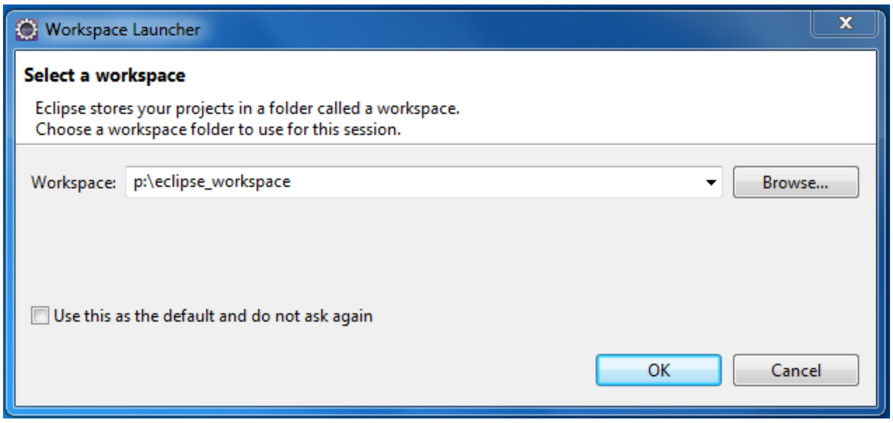
Computer Instructions

1 Making a screen-shot

Click on 'Start', type 'Snipping Tool' in the search field, press 'enter'. Use the tool.

2 Eclipse

Under the "Specialist Apps", open the folder "College of Science". Within this folder, open the folder "Computer Science". There, you find the program "Eclipse". When you start Eclipse you might be asked for the workspace path. This path should be set as follows:



Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: p:\eclipse_workspace

□ Use this as the default and do not ask again

2.1 Making a new project

1. Click `File` → `New` → `Java Project`.
2. Typing a good project name i.e. `Sphinx`.
3. Click `Finish`.

2.2 Importing a file into a project

1. Expand your project, say `Sphinx` in the left hand panel (Package Explorer),
2. Right click the `src` folder, click import.
3. Select `File System` under `General`, click `Next`.
4. Locate the directory containing the `Sphinx.java` file, click `OK`.
5. Check the file, e.g. `Sphinx.java`, in the right hand list, Click `Finish`.

2.3 Running a program

You run a program, e.g., `Sphinx.java`, by clicking the play icon. This may bring up a wizard where you need to select to run a `Java Application`. You may need to show the `Console` view by clicking `Window` → `Show View` → `Console`.

2.4 Activating JUnit4 for a project

1. Right-click on your project and select `Properties`.
2. Click on `Java Build Path`.
3. Select `Libraries`
4. Select `Add Library`.
5. Select `Junit`.
6. Click on next, select the Junit Version `JUnit 4`.
7. Click `Finish.`
8. Click `OK.`