CS265 Software Testing
Lab Week 05

Statement & Branch Coverage
(2.5%)

Exercise 1: Combinational Testing (1%)

Consider the method `bookSeatExtras` that has the following signature:

```
public static Status bookSeatExtras(TicketType type, boolean
sitTogether, boolean meal)
```

and the two variables Status and TicketType being defined as:

```
public static enum Status { ACCEPTED, UPGRADE, ERROR };
public static enum TicketType { STANDARD, PLUS };
```

This method determines if an extra can be selected for a flight given a specific type of ticket. The following conditions apply: With a standard ticket, any type of extra --- sitting together or the meal option, or both --- results in an upgrade request. With a plus ticket, one type of extra is accepted, but if both extras are requested, the system demands an upgrade. Independent of what type of ticket you have, if you do not request anything, the resulting Status will be an error.

Your colleague identified the following three causes for you:

- ticketType = STANDARD
- sitTogether
- meal

plus the effects:

- accepted
- upgrade
- error

Complete an initial truth table that includes all of these causes and its effects as given by your colleague.

Important note: When building truth tables, we apply a particular notation with respect to the order of true and false cases. For a full set of combinations (i.e. in an initial truth table), false (F) is situated on the left, and true (T) on the right of the table. We organise the creation of truth tables along half-way splits for false-true between each cause (please review lecture slides for examples). This method has been repeatedly demonstrated in class and prevents errors and confusion. Please stick to this notation for ALL exercises in today's lab, in future labs and in the exam.

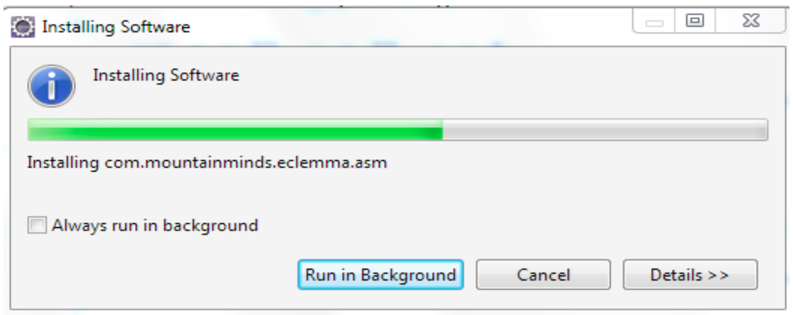
You must complete the Moodle quiz in order to collect the points.

Exercise 2: Basic SC/BC coverage with EclEmma (1.5%)

You will first need to install a plugin called *EclEmma*. Like TestNG, this only needs to be installed once is it is not already installed on your lab machine.

STEP 1 (Installation of EclEmma):

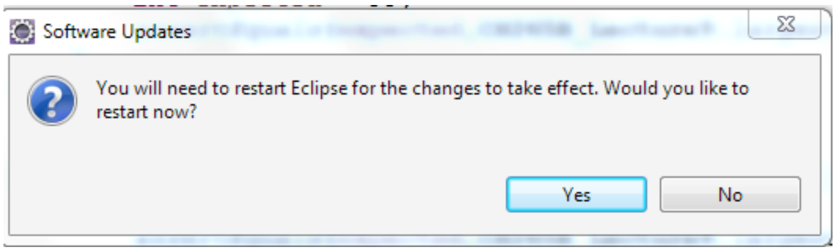
- Select the HELP menu, select MARKETPLACE and search for 'EclEmma'; Note: Your version may install packages differently: Find out how.
- Click the INSTALL button, accept the license, and install;
- When asked about certificates, accept to proceed;
- You may need to restart Eclipse to activate EclEmma.


Installing Software

Installing Software

Installing com.mountainminds.eclemma.asm

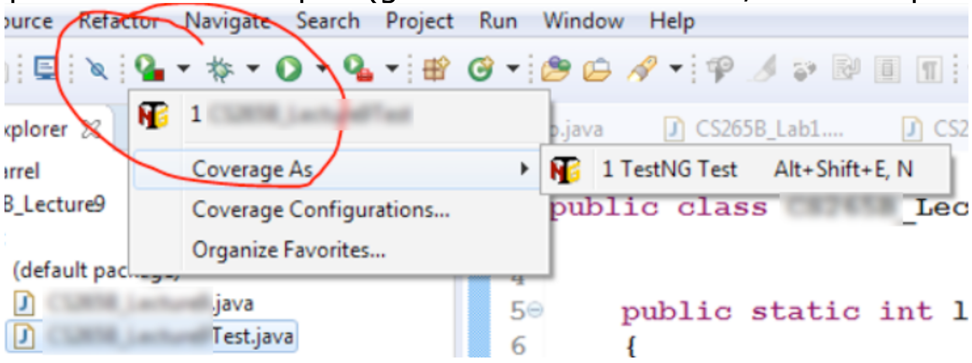
Always run in background


Software Updates

You will need to restart Eclipse for the changes to take effect. Would you like to restart now?

STEP 2 (Check if EclEmma is active):

- A successful installation should display a new button in your Java Perspective within Eclipse (green button with RED/GREEN square):



We are now ready to perform code coverage with EclEmma in Eclipse!

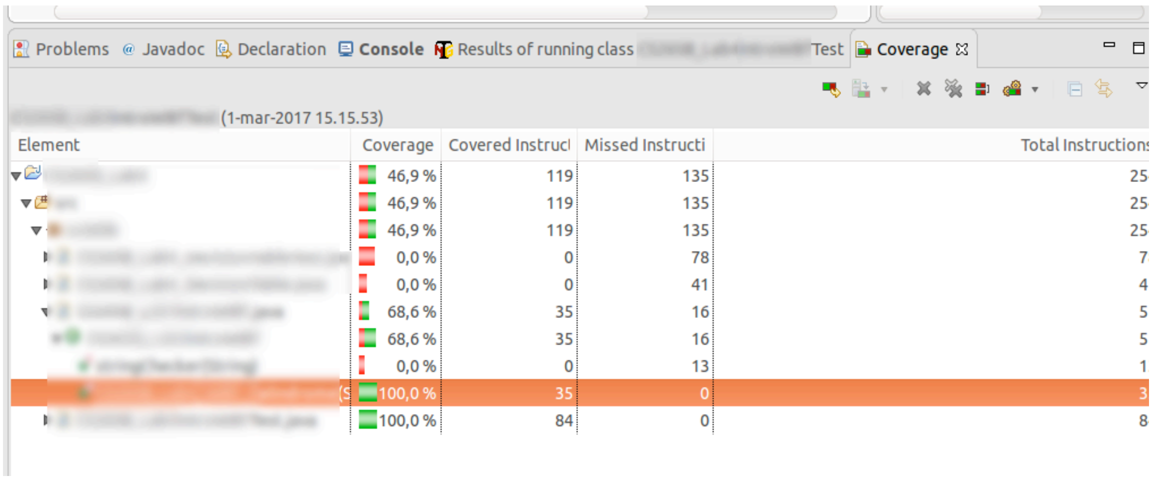
STEP 3 (Basic statement coverage):

- Consider `CS265_Lab_Week05_IntroWBT.java` as provided on Moodle;
- This source file contains <u>one</u> method called `isPalindrome`.

This method returns true if the string parameter inputString is a Palindrome. The inputString is NOT considered a palindrome if it contains any characters <u>except</u> [A-Z] (lower and upper case) and digits 0-9.

Perform a Statement and Branch Coverage Analysis of this method.

By considering the statements and their branches – you are required to develop TestNG tests for each of these conditions.



[redacted] (1-mar-2017 15.15.53)

Element	Coverage	Covered Instruc	Missed Instructi	Total Instruction
[redacted]	46,9 %	119	135	25
[redacted]	46,9 %	119	135	25
[redacted]	46,9 %	119	135	25
[redacted]	0,0 %	0	78	7
[redacted]	0,0 %	0	41	4
[redacted]	68,6 %	35	16	5
[redacted]	68,6 %	35	16	5
[redacted]	0,0 %	0	13	1
[redacted] (S	100,0 %	35	0	3
[redacted]	100,0 %	84	0	8

Note: To get coverage information in addition to your test results, you right-click on the TestNG file and choose *Coverage As -> TestNG Test*. This will then run all your Test NG tests plus calculate and visualise your achieved code coverage. To see the resulting code coverage, make sure you first select the class/file that you tested (CS265_Lab_Week05_IntroWBT.java). Click the *Coverage* tab at the bottom of the Eclipse window to get coverage information and make sure you are looking at the right scope of coverage (e.g. class vs method coverage). Clicking the *Results of running class* tab shows a list of results. Clicking the *Console* tab shows the raw result output of your test.

Tasks

- Write a sufficient number of TestNG tests to gain 100% statement and branch coverage for the method `isPalindrome`;
- Take a screenshot of your achieved coverage (from the 'Coverage' tab, as shown in the figure above) for the method in Eclipse;
- Upload this screenshot to the submission box that is provided with the lab. **You must do this to receive points for this exercise.**