Lab 8 Solutions

This lab was quite like a tutorial. There are very many correct answers. Take these solutions as a guide only.

Black Box Testing

This part was to test if you could come up with a suitable set of test cases.

Our code is supposed to take names which are between 3 and 10 letters long. We have three equivalence classes to check here. All names of length less than 3, all names between length 3 and 10, all names of length greater than 10. We could just test one name from each equivalence class (so "Bo", "Craig" and "Wolfeschlegelsteinhausenbergerdorff"). We could also test the boundaries of our classes. So a name of length 2, a name of length 3, a name of length 10 and a name of length 11 would be a sensible choice. Notice that not all errors regarding length would be caught by either of these options. This is deliberate, and is included to show the limitations of this type of testing.

Our code is also supposed to reject characters which aren't letters. This was included to
other sets
that do not c
If we wish to te
would suffice.

Code coverage, making sure the whole sys

It was expected that you would have chosen a few names to enter as input than the rather minimal set above. We hoped that you'd find that you could omit some of your choices without effecting code coverage. Also, if you're not hitting all the lines of code then we hoped you'd add some more names to your test set. Notice that lines like class declarations don't get hit by the code coverage tool.

Mutation testing

You were to fix some errors the code, in the Validator class, before creating mutants for testing the suitability of included junit tests (and more if you wish to include them). Line 9 should be changed to use regx, and the "!" should also be removed. The lines which deal with the length of the input both have errors. Line 12 should be removed completely.