# Project 4

*COSC 603 Software Testing*

*Sigrid Berkebile-Stoiser*

*4/6/2016*

## Task 1

The installation and use of the Eclipse Emma plugin was straightforward and did not cause any problems. I thought the coverage highlighting and comments marked by the little diamonds on the left border of the editor generated by Emma were very useful. By looking at the code that is marked up yellow or red, one gets immediately the idea what test cases are missing. The only thing that was a little confusing in the beginning was that the Junit test classes were analyzed for test coverage too, so the total project does not have a 100% coverage although all the relevant Java classes do. But after reading the Emma online documentation, I saw that that is a common question. I did a good job with my test cases in Project 3 and achieved 100% code coverage for VendingMachineItem.java and VendingMachine.java. Therefore, I did not have to add or change any of the existing test classes.

It was really hard for me to generate the test reports. The plugin worked fine, right after I installed it. However, I did not export the html files right away. When I tried to re-run the coverage test, I ran in a ton of problems and could not open the coverage view. I actually had to use a different computer to generate the reports, for reasons still unknown.

## Task 2

The CodePro analysis tool was easy to install and straightforward to use. It was very easy to generate the unit tests which covered over 80% of the code instantly. When looking through the generated test cases, I noticed that I had set up my own test cases quite differently and used more variations for the slot index and balances. However, that does not seem to make a difference for the code coverage. I am not sure if it is important, for example, to add items to all possible slots or if it is just enough to test out one. I also wondered how the test cases are generated because some of them contain the same code although they have different method names, testVendingMachine\_1() and testVendingMachine\_2(). I did not try to generate test cases with the test case table in the CodePro window as it was easy to write test cases for the uncovered code by hand. On the whole, the tool is extremely useful and able to save the tester hours of work!

The only thing I did not like was that it is not possible to minimize the window with the results what makes it hard to work on the code in the editor.