Atraya Mukherjee 1001144456
Raveena Jadhav 1000833967

# Summary and Discussion

In this project, we implemented Junit in Eclipse IDE to test a mid-size code. Using Eclipse was highly beneficial as Eclipse has Junit in-built in its build and run functionalities. We were able to cover 90+% of our source code Printtokens2 and 90+ % of our testcode Printtokens2Test. In Printtokens2 almost all the methods were sufficiently covered. However, few branches were missed in some methods. JaCoCo's visual representation of Test coverage greatly helps in improving testing methods.

The crux of test generation is creating CFGs using block tables. This is particularly highlighted while generating coverage. If CFGs are incorrect, the coverage is poor and test data does not add to quality assurance of the code. We were able to find numerous faults in the code due to proper test generation.

Overall, both Junit and JaCoCo are powerful tools that will be heavily utilized in the professional world. This project gave a practical viewpoint of testing and methodologies involved in it. Theoretical aspect of testing is very different to analyzing segments of code yet both have the same underpinning procedures. Now we see the true value aligned with test driven development especially in Java as we can implement it.