Deliverable 5

For this deliverable, we were to inject faults into our code base with creating at least five test cases to fail. Since we were testing our own code base, however, we had to once again work around the problem of injecting faults into our code base while still being able to test it using our code base.

To overcome this, we cloned the necessary Python files into a “brokenPythonFiles” directory. This directory could contain the fault-injected Python files. This eliminates the chance of mixing fault-injected files with non-fault-injected files. After this, we re-pointed the Driver to this directory to execute the test cases.

In practice, the user would not have to do this to this degree, just rewrite the test case to point to a fault-injected file as they specified. Our approach was only to overcome the complexity and challenge of testing our own code base.

We successfully injected faults into our code base to create seven test cases to fail. Thankfully, as we hoped, this did not break the entirety of the test cases. After injecting faults, we had sixteen test cases that were still successfully passing. Additionally, these faults did not break all of the test cases for that specific class. Of the original ten test cases we had for this particular class, we had three test cases still passing, and one of those was for this specific method.

Because of the work we put into the last deliverable, this deliverable proved to be less challenging, and we were able to inject these faults with greater ease than initially anticipated.