Nathan Tung
Com Sci 31 (Lec 2)
September 28, 2011

CS31 Project 1 Report

While the C++ program certainly compiles and produces correct results in most general cases, certain numerical inputs will produce unsatisfactory outputs (as in step 5). For example, if the input for `gasConsumed` is ever 0, then the double `mpg` is essentially dividing by 0. Thus the calculation results in "1.$" miles per gallon – a nonsensical answer. In another case, if either of the input doubles are negative for whatever reason, the calculation will also result in a negative miles per gallon, which is not possible. Although these inputs may not be realistic, the program should still be able to avoid producing invalid results.

In logic_error.cpp, I introduced several sources of possible human error, the first of which is multiplying `milesTraveled` by `gasConsumed` rather than dividing them. This would produce an incorrectly huge answer in almost all cases. Secondly, I changed the precision to 11, which would cause the program to spew out 10 more decimal places than usual. Neither of these errors are detected or accounted for by the program itself.

In compile_error.cpp, I first removed the semi-colon at the end of the declaring the double `gasConsumed`, which on compilation gives a "build error." I then used incorrect capitalization for the doubles `milestraveled` and `gasconsumed` in the calculation for `mpg`. Lastly, I misspelled "precision" as "precisoin." As it turns out, all three compilation errors result in a so-called "build error," to which Visual C++ will attempt to run the last successful build.