**Project 1 Report**

In the original program (original.cpp), I inputted a sets of integers that caused the program to output unusual outputs. I inputted that there were 4 Volkswagen owners, 6 owners who will take a payment and keep their car, and 8 owners who will return their car for a refund. The program outputted that 150.0% of the owners will keep their car and 200.0% of the owners will demand a refund. The integers are nonsensical because it should be impossible that 6 owners will keep their car and 8 owners will return their car when there are only 4 Volkswagen owners. Also, the sum of the percentages of those who will keep and return their car should equal 100%; however, the summation of 150% and 200% is 350%, which does not equal 100%.

In the program with the logic error (logic\_error.cpp), I changed the multiplication operator (\*) to an addition operator (+) in line 20. Just as the definition of a logic error stated, the program compiled correctly without any errors but produced a nonsensical output. I inputted that there were 5 Volkswagen owners, 3 owners who will take a payment and keep their car, and 2 owners who will return their car for a refund. However, the program outputted that 100.0% of the owners will keep their car and 40.0% of the owners will demand a refund. First of all, the two percentages should combine to be 100%, which would represent all 5 owners. However, 100% plus 40% equals 140%, not 100%. Secondly, because I inputted that 3 owners will keep their car, the program should output that 60% of the owners will demand a refund. But instead, it outputs that 100% will demand a refund, which is logically incorrect. As a result of the two percentages not adding up to be 100% and the percentage of owners who will demand a refund not being 60%, there is a logic error in the program.

In the program with the compilation error (compile\_error.cpp), I made two changes – removing a semicolon and a close parenthesis – that would represent mistakes someone might make. For the first change, I removed the semicolon from the end of line 9, representing the mistake of putting a semicolon at the end of a statement. As a result, it caused both numberSurveyed and numKeep to not be declared. Error messages that the compiler reported included that the compiler expected an initializer in line 10 and that numberSurveyed and numKeep were not declared in lines 14 and 16, respectively. For the second change, I removed the close parenthesis in the if statement in line 30. As a result, the condition in the if statement was not completed. The error message for the second change was that the compiler expected a close parenthesis before the “cout” in line 31. In the end, both of these changes caused the program to file to compile correctly.