Kevin Hahn

CS 31

UID: 504-914-505

Project 1 Report

In logic\_error.cpp, I introduced a logic error by switching the values of pctApprove and pctDisapprove. Now, when the code is run, the pctApprove that is printed is actually the percentage of people who disapprove and the pctDisapprove that is printed is actually the percentage of people who approve. Although the code still compiled, the outputted results will be clearly incorrect.

In compile\_error.cpp, I introduced 2 different compiler errors. My first compiler error introduced was a syntax error by removing the semicolons from the variables numSurveyed, numApprove, numDisapprove. Without the semicolons, these variables are not officially declared and the source code therefore cannot compile. The second error I introduced was misspelling the variable numSurveyed as numSurvey in the assignment of pctApprove. This caused an error because numSurvey is an undeclared identifier.

In original.cpp, inputting 0 for numSurveyed would result in pctApprove and pctDisapprove to be printed as “inf%” for nonnegative values of numApprove and numDisapprove, or “-inf%” for negative values of numApprove and numDisapprove. When numSurveyed is 0, also inputting 0 for numApprove causes pctApprove to be “-nan(ind)%” and inputting 0 for numDisapprove causes pctApprove to be “-nan(ind)%”.

For positive nonzero integer values assigned to numSurveyed, inputting positive numbers for numApprove and numDisapprove can still lead to nonsensical results. For example, if numSurveyed is 100, and numApprove is 60, and numDisapprove is 50, then the program still outputs the pctApprove and pctDisapprove as 60.0% and 50.0% even though the sum of these 2 proportions do not add up to 100.0%. Whenever numApprove and numDisapprove are equal to each other, then the program will output the conclusion that “More people disapprove than approve” even though this is clearly not true.

Assigning a negative, nonzero integer value to at least one variable out of numSurveyed, numApprove, and numDisapprove will produce nonsensical results regardless. For instance, if we input 100 for numSurveyed, -60 for numApprove, and -40 for numDisapprove, then pctApprove and pctDisapprove would have values of -60.0% and -40.0%. Even though these results are clearly nonsensical because nonzero negative integer values are assigned to these 3 variables, the division in computing pctApprove and pctDisapprove still produces mathematically correct results with the correct signs.