CS31 Assignment 1

For step 5, I started by inputting various combinations of positive and negative integers to determine how the program would respond. More specifically, I said -26 people took the survey and -13 approved while -13 disapproved. I received an output saying 50% approved and 50% disapproved which would make sense normally, but it could not realize a negative number of people did not make sense to begin with. It also said more people disapprove than approve despite the numbers, even though negative, were technically the same value. I then said 100 people responded to the survey and said 0 agreed and 0 disagreed. The percentages were technically accurate at 0% each, but it again favored the “more disapprove than approve despite the responses being the same. I can conclude from this that the original program cannot interpret any negative integers as any form of input, and it cannot say, “the same number of people approve as disapprove”.

For step 6, I added a zero to the 100 in line 21 so the calculation that outputs what percentage of people disapproves, despite it being a simple typo, is 10 times what it should be. I also reversed the greater than sign (>) to be a less than sign (<) without switching the order of outputs so the program will confuse which group (approve or disapprove) had more respondents. In doing so, the program builds correctly but produces unrealistic and incorrect results from completely normal inputs.

For step 7, I needed to create bugs in the actual language and not just calculations to cause the program to fail to build correctly. I started by writing “Namespace” instead of “namespace” in line 5 which caused all of the standard language phrases such as cin or cout to be unrecognized. This alone was enough for the program to fail, but I explored further by removing the semicolons after calling the variables in lines 9,10, and 11. This basically made these 3 lines void thus highlighting the importance of the semicolon. I also changed “cin” to “cins” on line 16 which, even though I already messed up the directory by changing namespace to Namespace, would have been enough on its own to cause problems. Finally, debugging the program yields the message, “There were build errors. Would you like to continue and run the last successful build?” proving the small bugs I introduced were enough to keep it from building at all.

When tested on g++ on the Linux servers, the first two programs yielded the exact same results with the same input cases. The compile\_error program produced the output,

“[august@lnxsrv07 ~]$ ./compile\_error.cpp

./compile\_error.cpp: line 1: //: Is a directory

./compile\_error.cpp: line 2: //: Is a directory

./compile\_error.cpp: line 3: $'\r': command not found

./compile\_error.cpp: line 5: using: command not found

./compile\_error.cpp: line 5: //: Is a directory

./compile\_error.cpp: line 6: $'\r': command not found

./compile\_error.cpp: line 7: syntax error near unexpected token `('

'/compile\_error.cpp: line 7: `int main()”

This means the program was unable to compile on g++ as well, but it did identify I had an error on line 5 using an unrecognized command. This assignment taught me the importance of accuracy in my typing and formatting while programming, how to run g++ on the Linux servers, and how that may help me to identify various problems with my future programs.