Maeneka Grewal

CS31

Project 1

**STEP 5:**

If you input ‘0’ for the number of people surveyed, the number of people who approve, and for the number of people who disapprove, it returns ‘nan%’ for the approval and disapproval percentages, as well as stating that more people disapprove than approve which is false.

If you input ‘0’ for the number of people surveyed, and integers greater than 0 for either the number of people who approve or disapprove, the computer will return ‘inf%’ for their respective percentages because of the division by zero that occurs in the program.

If you input an integer greater than the number of people surveyed for either the number of people who approve or disapprove, the program returns a percentage over 100 instead of catching the error that it is in fact impossible for more than 100% of the people surveyed to approve or disapprove. For example, if you input that 5 people are surveyed, 10 people approve, and 15 people disapprove, the program will output that 200% approve and 300% disapprove.

If the integers inputted for the number of people who approve and disapprove are the same, the program will always state that more people disapprove than approve although that is false because the program does not check to see if the values are equal.

**STEP 6:**

The only change made to the source code was in lines 20 and 21, when calculating the double variables ‘pctApprove’ and ‘pctDisapprove’. Instead of dividing the number of people who approve or disapprove by the number of people surveyed, the number of people who were surveyed was divided by the number of people who approve or disapprove. This resulted in a program that compiled, but that outputted the wrong percentages, almost always over 100%. For example, if 10 people were surveyed and 6 people approved and 4 people disapproved, the program outputted that 250.0% approved and 166.7% disapproved, instead of the correct answers of 60% approving and 40% disapproving.

**STEP 7:**

The two errors in the source code introduced can be found in lines 9 and 13. In line 9, the variable ‘numSurveyed’ was declared as a String instead of an int. The program cannot perform arithmetic operations with a String and therefore failed to compile. The specific error messages were “Build Failed”, as well as red notes on lines 20 and 21 that said doubles and strings were invalid operands in a binary expression. The error in line 13 is simple but common, a forgotten semi-colon. The program does not recognize the end of a code expression and cannot compile. The specific error messages were “Build Failed” and a red note on line 13 that said a ‘;’ was expected after the expression. The g++ compiler also displayed messages that the operation couldn’t be performed because of a missing semicolon and the mismatched data types.