Raj Kumar

Professor Smallberg

Computer Science 31

12 October 2020

Project 1 Report

In Step 5, I used negative numbers inputs to make the program output negative percent chances for Joe and Donald’s chances of winning, which is nonsensical. I also used inputs in which the total number of voters surveyed was less than the sum of the people that said they would vote for Joe and Donald, which then could cause the proportion of people saying they would vote for Joe or Donald to be higher than 100%, or sum up to higher than 100%, which also doesn’t make sense. If Joe and Donald have the same number of people saying they will vote for them, then the program would predict Donald will win the election, which would be incorrect. In this case it should say inconclusive or something along those lines.

In Step 6, I swapped the inputs forJoe and forDonald that the user gives when answering the questions about how many will vote for them. In other words, the amount of people saying they were voting for Joe would be saved under the forDonald variable and the number of people saying they were voting for Donald would be saved under the forJoe variable. This would then swap the percentanges that correspond to how many percent of people say they will vote for Joe and Donald, and would also predict the election winner wrong (when Joe should be predicted to win, the program would say Donald is predicted to win and vice versa).

In Step 7, I created a compile error by erasing a double quotation character that looks like this: “. I erased this character after the question mark in the code line 13:

cout << "How many registered voters were surveyed? ;

I picked this error because this was a common error for me when I was learning Java and thus would be a realistic error I would make while learning C++. When compiling, 4 different errors were shown. The first error shown had the error code E0008, which said “missing closing quote.” This was the expected error as that was exactly what I had erased. The second error shown had the error code E0065, which said “expected a ‘;’.” This occurred because the semicolon at the end of the line of code was now being treated like it was part of the string due to the missing end quotation marks that would mark the end of the string. The third error shown had the error code C2001 which said “newline in constant,” which occurred because the starting in line 13 no longer had a marked end, causing visual studio to say that I can not have the starting of line 14 as part of my string in line 13, as that is not allowed in C++ without putting a backslash in the end of line 13. The fourth error shown had the error code C2146 which said “syntax error: missing ‘;’ before identifier ‘cin’.“ This error appeared because there was no semicolon before another identifier was in the code, which isn’t allowed in C++ syntax. The semicolon is still there, but since there was no longer a marked end of the string, visual studio thinks that the string includes the semicolon and thus the semicolon is no longer serving its purpose of marking the end of the line of code.

In Step 7, the second compile error I created was in line 27 of the code, where I omitted a ‘<’ character. Line 27 appeared like this:

cout << pctJoe << "% say they will vote for Joe." < endl;

This only created two errors, where the first error had the error code E0349, which said “no operator “<” matches these operands.” This occurs because Visual Studio rightfully recognizes that the ‘<’ doesn’t match the situation, which instead requires ‘<<’.

The second error had the error code C2676 and said “binary '<': 'std::basic\_ostream<char,std::char\_traits<char>>' does not define this operator or a conversion to a type acceptable to the predefined operator.” With my limited knowledge of C++, it seems that this is because the ‘<’ was not one of the valid options Visual Studio expected to be there in that line, expecting something like the ’<<’ that was supposed to be there.