Jason Tran

CS31 Project 1

10/3/22

For the original file (original.cpp), that I directly copy and pasted and didn’t alter in any way. I *put* for my first try basic inputs:

**How many registered voters were surveyed?** *100*

**How many of them say they will vote for Gavin?** *90*

**How many of them say they will vote for Brian?** *10*

The result was:

**90.0% say they will vote for Gavin.**

**10.0% say they will vote for Brian.**

**Gavin is predicted to win the election.**

These results were expected because if 90 out of 100 people voted for Gavin, that means 90% voted for him, which is the winning majority.

The illogical inputs, for step 5, I entered into the original program was based off of having people voting for Gavin and Ben, despite the total numbers of registered voters being 0. It produced these results:

**How many registered voters were surveyed?** *0*

**How many of them say they will vote for Gavin?** *90*

**How many of them say they will vote for Brian?** *10*

**inf% say they will vote for Gavin.**

**inf% say they will vote for Brian.**

**Gavin is predicted to win the election.**

As expected, this would lead to weird results due to the code dividing by 0, which is undefined in the world of Mathematics. Another illogical inputs I tried was entering more voters for Gavin (110), despite the total number of registered voters (100) being less overall. The code outputted 110% say they will vote for Gavin, which is nonsensical because usually the cap should be at 100%. Also putting in 10 for Ben produced 10%, which also shows how the program treats the percentages independently from Gavin.

For step 6, I removed multiplying by 100 in the variables pctGavin and pctBrian. The program ran successfully, but the results were not correct nor the same to the ones I put in the original file. The results were:

**How many registered voters were surveyed?** *100*

**How many of them say they will vote for Gavin?** *90*

**How many of them say they will vote for Brian?** *10*

**0.0% say they will vote for Gavin.**

**0.0% say they will vote for Brian.**

**Gavin is predicted to win the election.**

This is definitely wrong since if 90 out of 100 voted for Gavin than it should be 90% for Gavin, like in the original.cpp.

For step 7, one mistake I introduced was removing or “forgetting” to put in #include <iostream>. This lead to a multitude of compilation errors. An example of one from the terminal was: **1>E:\CS31\Project1 10-4-22\compile\_error.cpp(13,2): error C2065: 'cout': undeclared identifier**.

There were also more so called “undeclared identifier” errors came right after this line, which leads me to believe that #include <iostream> was important to defining what some useful console tools do like cout and cin.

Another mistake I introduced was taking out a semicolon after declaring the variable forBrian. This lead to this error: **1>E:\CS31\Project1 10-4-22\compile\_error.cpp(13,2): error C2146: syntax error: missing ';' before identifier 'cout'.** This is interesting because the compiler tells you what and where the compilation mistake was and is pretty useful because forgetting a semicolon is what some novice programmers do.