Daniel O’Laughlin

CS 31

Project #1 Report

In step 5, I inputted “500” for the integer variable “numberSurveyed” but inputted “2000” for the integer variable “forJerry” and “300” for the integer variable “forNeel”. This means that more people said they would vote for Jerry than the number of total people polled which makes no sense. As a result, the program produced results that suggest that 400% of people said they would vote for Jerry while only 60% of people said they would vote for Neel. These results exceed 100% and are nonsensical.

In logic\_error.cpp, the program produces incorrect results from reasonable input because in the following line of code, there is an extra 0 to the left of the decimal place creating the number “1000.” rather than the intended “100.” This results in the program compiling, but producing the wrong percentage values because by definition, percentage means parts per one hundred, not parts per one thousand. So, as you can see there are multiple ways to create outputs containing nonsensical percentages.

double pctJerry = 1000. \* forJerry / numberSurveyed;

In compile\_error.cpp, an error message is produced that states, “Expected ‘;’ at end of declaration”. This is because a semicolon was omitted after the declaration of the integer variable “forNeel”. A semicolon is required at the end of this declaration to signal the end of a statement but not the end of the line. This could allow you to have multiple statements on a single line. In this way, semicolons allow you to have more control over the formatting of your project. Also, the compiler needs to know when the statement is finished, so without a semicolon the program will not compile.

The other error in the code lies in the line that states “double pctJerry = 100.0 \* forJerry / numbersurveyed;”. In this line, an undeclared variable is present. The error message produced is as follows, “Use of undeclared identifier ‘numbersurveyed’; did you mean ‘numberSurveyed’?” This undeclared identifier is “numbersurveyed”. Although it seems as if it was previously declared in the program, the previous declaration was written as “numberSurveyed.” This failed capitalization of the “s” is crucial because without this capitalization, it will not be recognized as the same variable.