Ethan Duong

ID: 805124044

Step 5:

Suppose we run the original code with the following inputs: There are 100 COVID-19 tests with 120 being positive and -20 being negative. There are 130 false positive and 10 false negative tests. Already we can see that these inputs are nonsensical for a multitude of reasons (for one, it is impossible to have 120 out of 100 tests be positive).

The output of the code is as follows: -8.3% of the 120 positive tests were actually positive. 150.0% of the -20 negative tests were actually negative.

We can see this output also does not make sense.

The code catches some illogical data inputs. Specifically, it catches when the number of positive and negative tests do not add up to the total number of tests and notifies the user using a print statement. However, using the inputs above, we are able to bypass this check as 120 plus -20 tests do end up adding to 100.

Step 6:

The original code:

double pctPos = (1.0 \* positives - falsePositives) / positives \* 100.0;

The changed code that creates a logic error:

double pctPos = (1 \* positives - falsePositives) / positives \* 100;

Let the number of positive tests be 100 and the number of false positives be 50 (these are logical inputs). Then in the process of doing the calculation for the second line of code, we divide 50 by 100 where both 50 and 100 are integers. The result is then truncated; instead of being 0.5 it is 0. We then multiply 0 by 100 and end up with 0. Thus the code outputs that 0% of the 100 positive tests be actually positive. The original code converts 50 to a double before dividing by 100 thus avoiding truncation entirely.

A second logic error occurs if we declare pctNeg as an int instead of a double. This causes unwanted truncation after the percent is calculated (ie. 65.9% could become just 65%).

Step 7:

We introduce a build error by removing a semi-colon and removing the type declaration for variable pctPos. We are given error messages that complain about a missing semi-colon as well as an undeclared identifier named pctPos.