Project 2 Report

1. A brief description of the notable obstacles I overcame.
   1. Initially, I had trouble coding the correct output precision, as I don’t know how to round the number to three digits after the decimal point, and I wasted a lot of time trying to round the find manually. Then I suddenly recalled in Project 1, the example code used ‘cout.precision(1);’ to round the output to one digit after the decimal point. After experimenting to implement this in my code, I realized that I can use ‘cout.precision(3);’ to round the fine to three digits after the decimal point.
   2. When I finished coding my program after I read the project spec the first time, I found that the calculated fines for some input values do not match the correct fine calculated by the Financial Fraud Calculator. After reading the project spec a few more times, I realized that I didn’t notice the fine for ‘florida’ and ‘new york’ for the next 90 million USD after the first 60 million USD is 18%, not 15.3%. After realizing my mistake, I fixed my code quickly and got the desired output. I learnt that I must be very careful reading the project spec, and I should not miss every single line in the project spec.
   3. I also struggled to figure out how to terminate a program in C++ before it reaches the last line of code. I was thinking of if there is an equivalent of the Python ‘break’ in C++. After reading the Project FAQ, I learnt that I can ‘return’ a non-zero value anywhere in my program if my program catches an error and needs to be terminated. So, I added ‘return 1;’ statements in my program when it catches the user inputs invalid text/integers.
2. Test Cases
   1. Empty strings input to test if the program recognizes blank user input, and displays the correct error message.
      1. (‘’, 100, ‘florida’)
      2. (‘A great house’,100, ‘’)
   2. Boundary case test to test if the program rejects input that are **not greater than 0**.
      1. (‘National Doral Miami’, -100, ‘florida’)
      2. (‘National Doral Miami’, 0, ‘florida’)
      3. (‘National Doral Miami’, 1, ‘florida’)
   3. Different location inputs to test if the program recognizes ‘florida’ or ‘new york’ and apply a different fine rate.
      1. (‘National Doral Miami’, 80, ‘florida’)
      2. (‘National Doral Miami’, 80, ‘new york’)
      3. (‘National Doral Miami’, 80, ‘California’)
   4. Numbers with decimal to test if the program accepts input that is not an integer. (‘A great house at Beverly Hills’, 123.456, ‘California’)
   5. Valid input that will not be rejected and check with the financial fraud calculator to see if calculations are correct. (‘A cheap property from USC’, 1, ‘California’)