Project Name: Project 2

Author: Marc Luzuriaga

UID: 205916585

**Obstacles in Project 2**

One obstacle that I encountered while doing Project 2 was trying to figure out the logic behind identifying free tax dates. I struggled with seeing how I could use the if statements to catch the dates. Although I struggled at first, I eventually found that you could nest another if statement within the state names if statement to catch the free tax date. And, within that if-statement, you would set the stateTaxRate = 0, avgLocalTaxRate = 0, and the levy = 0.

Another obstacle that I encountered was creating the error handles for the program. At first, I tried to catch the invalid input within the condition of the if-statement. However, this became difficult to debug. Instead, I used the condition of the if statement to catch valid input, and I would use the else statement to catch the invalid input.

**Test Data**

| State Name: | New Hampshire |
| --- | --- |
| Purchase amount: | 50 |
| Provide the month: | August |
| Provide the day: | 0 |
| Provide the year: | 200 |
| Invalid day! | |

Explanation: The reason why I used this test data is because I wanted to ensure that my program caught the error for valid days. I used the extreme of 0 because I wanted to know the extent to which my program caught the error. This checked that the day was equal to or above 1. In this case, the program did this.

| State Name: | Iowa |
| --- | --- |
| Purchase amount: | 50 |
| Provide the month: | August |
| Provide the day: | 5 |
| Provide the year: | 200 |
| Please pay a total of $50.00 | |

Explanation: There are two reasons why I used this test data to test my program. The first reason was because I wanted to test if the syntax and logic of catching the errors were correct. The second reason was because I wanted to test if the program displayed the decimal 0s in the output. In this case, the program did just this.

| State Name: | California |
| --- | --- |
| Purchase amount: | 50 |
| Provide the month: | January |
| Provide the day: | 5 |
| Provide the year: | 2020 |
| Invalid state! | |

Explanation: Although the inputs look valid in this program, I put a space after the word “California” when inputting the state name. The purpose of this program was to check if my program caught spaces after the string when catching invalid input. In this case, the program was able to catch this and output “Invalid state!”

| State Name: | Wyoming |
| --- | --- |
| Purchase amount: | 50 |
| Provide the month: | March |
| Provide the day: | 18 |
| Provide the year: | 2025 |
| Please pay a total of $52.68 | |

Explanation: I used this test case in order to see if my program was able to identify that 2025 was a valid input. In this case, my program was able to catch it because of the if statement that had the condition of “<=2025.”

| State Name: | Texas |
| --- | --- |
| Purchase amount: | ajsfjsafdj |
| Provide the month: | January |
| Provide the day: | 20 |
| Provide the year: | 42 |
| Invalid amount! | |

Explanation: I used this case because I wanted to check that I cleared the keyboard buffer, and the strings that were inputted were converted into a 0. Because it was converted into a 0, my program was able to check and identify that it was less than 1, which caught the invalid amount.