Programming Project Report: Project 2

Name: Riley Nadwodny

Date: 9/25/19

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement**

The goal of this programming assignment was to give us some more experience using input/output commands, as well as experience with conditional statements in C++. The goal of the program itself was to receive scores on a scale from 1-100 for a customer’s likeness of food and the customers budget, compare the scores, making food decisions for the customer from the chef, and then outputting their food, based on their score and budget inputs. Error handling required making sure that if a customers score was under 1, their score was set to 1 and if it was over 100, the score was set to 100, as those were the bounds for the scale.

**Design**

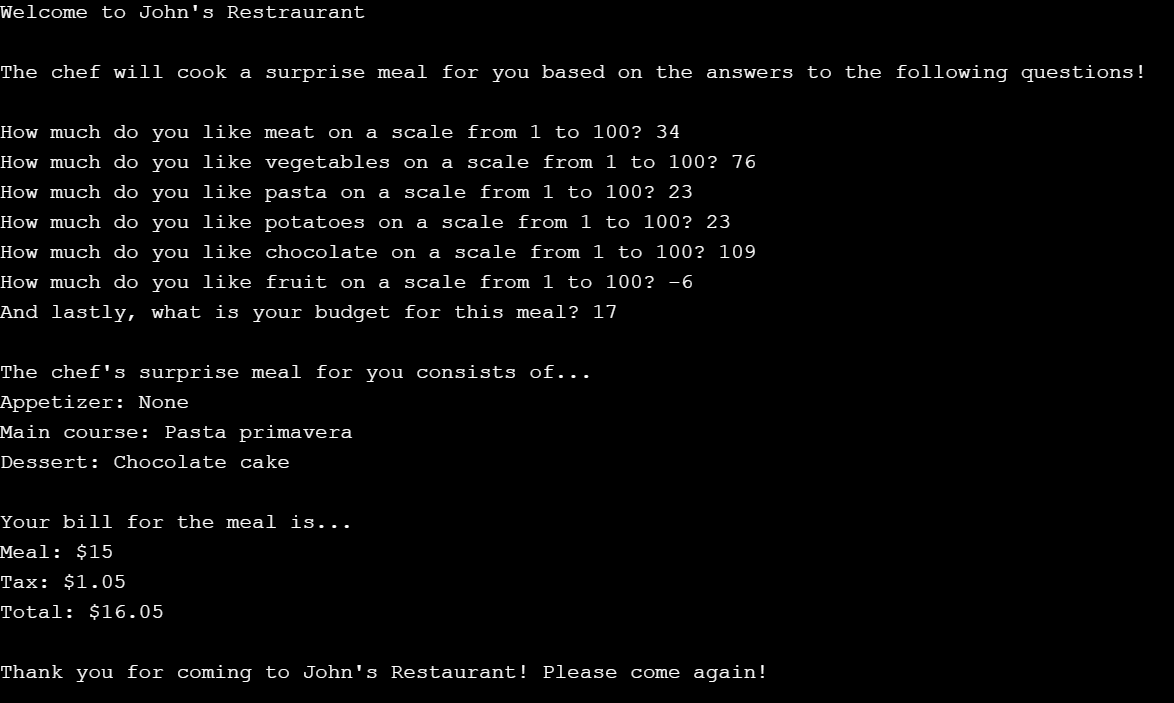
For data structures in this program, I used int, double, and string data types to receive inputs, be updated with values, and output values to the user. In this project, I used if else statements with nested else if statements inside of those. If the user had a budget over $20, they were sent to one if statement, and inside of there, their scores were compared with one another in a series of else if statements to determine what meal to give them, and what they could afford to get. If they had exactly $15 or between $15 and $20, they were directed to another else if statement that had nested else-if statements inside of it to compare their scores, and there is one last else-if statement for if the customer has exactly $10 or between $10 and $15. If their score was over 100, their score had to be set to 100 to stay within the upper bound of the scale. In the same case, if their score was lower than 1, their score had to be set to 1 in order to stay within the lower bound of the scale.

There were some pros and cons to this implementation in my opinion. A big pro for me is that I was able to do many statements to compare the customer score and make sure they got the right item, and that item would be input into the string value, which would be output at the end to give the customer their meal. The biggest con for me is that I believe using so many else-if statements in the way that I did made my code longer than it needed to be; I’m sure that I could have implemented a better design for this program, but I wasn’t too worried about it this time around.

**Implementation**

For this program, I did not start with any sample code around. I began by declaring variables to 0 or nothing in the case of strings, receiving inputs from the user and then outputting those back out, making sure that they had the right inputs. I then began making comments for the outer else-if statements, and then after doing that, I came up with more else-if statements within those in order to decide what meal to give to the customer. I wrote the first if block of the outer else-if statements, and then developed the inner else-if statements inside of there. I tested the whole block and then repeated the blocks of code with different condition statements for each one. In order to write this program from start to finish, I estimate it roughly took me about 5-6 hours to make sure all the cases I had thought of accounted for and implemented into the program in some way.

**Testing**

 For testing, I tried a variety of different inputs. For a lot of the testing cases, I input differing numbers for the scales and made sure they came out with the correct outputs. I also tested what the output would be if all of the inputs were the same number except for the budget, as well as what the output was if all of the inputs were either all above the upper bounds, lower than the lower bounds, or mixed between the lower and upper bounds. Not everything worked as expected at first, but after some code-checking and corrections, everything worked as expected in all cases.

**Conclusions**

Overall, I am satisfied with how this project turned out. At one point I was a bit annoyed when the outputs weren’t working correctly, but after taking a break and coming back to it, I saw what was wrong and was able to fix the code to have my program output the correct values. One thing I would definitely do differently next time is figure out how to make my code easier to read and cut a bit of the length of it off. I’m sure I could find a more efficient way to implement the conditional statements to reduce the lines of code. After all said and done, this project took roughly about 4 days to code and error-check.