Quest 5 (Key Quest)

100 EXP

CPSC121 SI

Craig: Welcome Back Programmer.

Craig: I'm sorry for missing your quest 4, but I heard that Rex filled you in.

Craig: Since you know about operators and variables, let's talk about constant variables and stream manipulators.

Craig: Constant variables cannot be changed during program execution. So if you can't change a variable, why use it?

Craig: The use of a constant variable can simplify program maintenance. For example, if your program has many calculations that utilize the tax rate, it would be beneficial to have those calculations use a constant variable called TAX_RATE instead of hard coding the value into every calculation.

Craig: This simplifies program maintenance because if the tax rate ever changes, you only have to change your constant variable's value instead of fixing every formula in the program!

Craig: Another note on constant variables that you probably already noticed was that constant variables are all caps! This tells anyone who looks at your code that you are using a constant variable.

Craig: Here are a few examples of how to declare constant variables:

```
const double TAX_RATE = .0975;
const int NUM_STATES = 50;
```

Craig: There's also the C-style way of naming constants:

```
#define NUM_STATES 50
#define PI 3.14159
```

Craig: This style of naming constants replaces every occurrence of NUM_STATES with 50, similar to find and replace. Notice how there is no semicolon? That is because if you put a semicolon, NUM_STATES would be replaced with 50; thus causing syntax errors all over our program (due to random semicolons).

Craig: Now, you must be wondering what a stream manipulator is. A stream manipulator is used to control features of an output field.

Craig: The four stream manipulators that we will be focusing on are setw(x), setprecision(x), fixed, and showpoint. In order to use these manipulators, you must include the iomanip library header!

Craig: You include this just like iostream. >>> #include <iomanip>

Craig: setw(x) is used to print a field of at least x spaces wide. You can control the orientation of the output in the field by using left or right as follows:

```
cout << setw(5) << left << sampleVariable << endl;
cout << setw(2) << right << sampleVariable << endl;
```

Craig: If the contents of the output is longer than the field, the field will stretch to accommodate it.

Craig: setprecision(x) is used to print a floating-point values using x significant digits. For example:

```
double sample = 132.4562;
cout << setprecision(3) << endl;
cout << setprecision(2) << endl;
cout << setprecision(5) << endl;
```

```
Output:
132
132e+002
132.46
```

Craig: If you combine fixed with setprecision, this ensures that the number will always show up as a decimal number, and the setprecision determines how many spots after the decimal point you go to.

```
double sample = 132.4562;
cout << fixed << setprecision(2) << sample << endl;
cout << fixed << setprecision(5) << sample << endl;
cout << fixed << setprecision(0) << sample << endl;
```

```
Output:
132.46
132.45620
132
```

Craig: Lastly, there is showpoint. This is used so that the decimal point is always shown, hence the name showpoint. You will use this for outputs such as prices because you want to show trailing zeroes to represent cents.

```
cout << setprecision(2) << fixed << showpoint << sample << endl;
cout << setprecision(0) << fixed << showpoint << sample << endl;
```

```
Output:
132.46
132.00
```

Craig: These stream manipulators are very useful when you are formatting your outputs and rounded to a specific precision.

Craig: Now, my boss needs a program that calculates an order made by a user. You will need 4 constants (tax rate = .0975%, burger cost = $3.50, fry cost = $1.25, drink cost = $1.00). My boss is opening up a very small burger hut that only has one kind of burger, one size fry, and one size drink…

Craig: The program must ask the user how many of each item they want, then total up the sub-total. Your program must display the subtotal, the calculated tax, and the total (after tax). Since the smallest unit is cents, please make sure the subtotal, tax, and total round to the nearest cent. Once you have your program working properly, make sure that if my boss ever changes his item pricing, all you have to do to update the program is change the constant variables' values.

Craig: Best of luck! Please show your SI leader your program once you are done.