Learning module 1

**Part 1:**

**Question 1:**

1. I changed my cout to “cout << “Hi my name is Aaron!” << endl;
   1. This printed “Hi my name is Aaron!” when I ran it.
2. I made my cout message “cout << “a;lsdjf lsdjfal; sdjf” << endl;
   1. This printed “a;lsdjf lsdjfal; sdjf” in the terminal when I ran it.
3. I changed the cout message to “cout << “According to all known laws of aviation, there is no way a bee should be able to fly.” << endl;
   1. This printed “According to all known laws of aviation, there is no way a bee should be able to fly.” In the terminal.
4. I made the cout message “cout << “this” << endl << “is” << endl << “sparta” << endl;
   1. This message printed out:

this

is

sparta

**Question 2:**

I made these lines of code:

cout << “Hey man” << endl;

cout << “what are” << endl;

cout << “you up” << endl;

cout << “to today?” << endl;

To output:

Hey man

what are

you up

to today?

This was to see what using multiple couts would do.

I also made this code:

cout << “hi” << endl << “how” << endl << “are” << “you?” << endl;

To output:

hi how are you?

I did this to try fitting multiple messages into one cout command.

**Question 3:**

The code I used is:

#include <iostream>

using namespace std;

int main() {

cout << "CS171 is the best.\n - Unless there are cookies\n - Or unless we are watching Star Wars\nOh! Star Wars movie night! with cookies!" << endl;

return 0;

}

**Part 2:**

**Question 1:**

There is not a modulus operation for floats because there is no need for it. The point of modulus is to give a remainder for an integer division, and since floats can return a precise decimal, there is no need to display a remainder. I did try to compile it, but before I even compiled it was highlighted in red and I was told that the operation was “invalid operands to binary expression.”

**Question 2:**

I used an inline function to output the operations. The strength of using an inline function to output operations is that it is less visual clutter on the screen. Along with that, it takes up less space in system memory because there are less variables being initialized. The strength of assigning outputs to variables, however, is that the cout is much neater and the calculations are done beforehand.

**Part 3:**

**Question 1:**

This program works rather simply. First, the total amount of the check is initialized as a float variable, since checks almost always have a decimal. After that, the number of people in the party is initialized as an integer variable, since there can’t be 1.5 people.

After initialization of these two variables, the user inputs the values for both the check and the number of people. As stated before, the check can have decimals and the number of people cannot.

Once these values have been assigned, the base amount each person has to pay if the bill is split is assigned to a float variable. This amount is the bill divided by the number of people. That value is then printed to the terminal.

Now the tip is calculated. This is done by printing “The amount each person should pay, plus \_\_% tip, is:” and then displaying the split variable times the tip percentage. The tip percentage for 10% tip would, for example, be 1.1. After that, the return key is pressed so that the output isn’t all on one line. This is done four separate times, once for 10% tip, once for 15% tip, once for 20% tip, and finally for 25% tip. That’s all there is to this program.

**Question 2:**

If someone enters 2.5 for the number of people paying, the compiler just drops what was after the 2 because totalPeople was initialized as a variable and there isn’t enough space in storage for the extra decimals. Having 2.5 people doesn’t make sense because half of a person can’t walk into an Applebee’s on a Friday evening and ask for a beer. Where would the liquid go? If they were split in half at the legs the liquid would simply not be consumed. If that person was split in half down the center (head to toe) the beer would just fall out of his body because there is nothing keeping it in. If the person was split diagonally, he might be able to get some of the liquid into his stomach, but one little problem would be that if a person is split in any which way, they are more than likely dead.

One way of dealing with this issue of assigning a floating point number to an int variable is by writing an if statement to essentially say “if totalPeople is not an integer, then spit out an error message and ask for input again”.