

1. Goals for today: Practice cout, cin and using strings. Learn comments! Learn new types: int and double. Explore assignment and try a slightly larger program!
2. Remember this sample program:

```
string your_name; //this is a comment, it is not seen by the compiler
cin >> your_name; //read in a value for the bucket, from the user
cout << "Your name is " << your_name << endl; //tell the user something
```

Too many comments can actually make the code harder to read. Use them for useful explanations or tricky code. Not for every line.

3. Does it have any style problems? Does it have any compiler errors? What does it do? What do we think it should do? Does it have any logical errors?
4. Make string variables for first names and last names. Read one of each in from the console. Output to the console the last name, followed by a comma, followed by the first name.

5. When we make a variable, this is called declaring the variable. This is called a declaration because we are declaring (to the compiler) that we want a bucket with a certain name and a certain type. It is important to keep that vocabulary word in your mind.
6. To make an integer variable, we can use the type `int`. Otherwise, we can attach it to output and input from the console just like we did with strings. Give that a try. Make a new integer variable, read in the variable from the console and then output that variable back to the console.

7. We can perform arithmetic on variables too. An example follows:

```
int velocity = 33;
cout << velocity * 2 << endl;
cout << "The velocity is now: " << velocity << endl;
```

8. What do you think the console output of the above statement might be?

9. If you need a numeric type with decimals, we will use `double`. As with integers, this has limitations on the range of real numbers that it can represent. We will not focus or reach those limits often, but it is important to know that they do exist.

10. It is important to understand how we, as programmers, can change the value of a variable (bucket). We have shown how the user might change the value of a bucket, but we have not explicitly explored that as programmers. We do that with something called an assignment. Assignments use the equals sign (`=`). There was actually an example earlier. We can set the value of a variable as a programmer when we declare the variable like this:

```
string prof_name = "Ramsey";
```

11. Or we can change the value of an existing variable later with an assignment like this:
- ```
prof_name = "Dr. Ramsey";
```

12. If we imagine that the velocity variable above was already declared, we might increase our velocity by doing something like this:

```
velocity = velocity + 1;
```

13. Remember, to use an assignment, that variable must have already been declared. In addition, this assignment happens just the one time. It is not an eternal connection between the variable and the assigned value. You should not think of them like chemical equations or even mathematical equations. The line above would be broken if you did that. It is better to think of it as: "let the value of this bucket be this value right now".

