# CSI 201
# Function Definitions

1. We're going to be learning how to write our own functions. This is an immensely powerful tool that helps us in both modularity and code re-usability. Because modularity often leads to pieces that are easier to test and debug, this will be a big boon once we completely comprehend them. Let's start with a a few prototypes.

2. When we use the #include operation, we're including a bunch of prototypes for functions that already exist. We might be including some other things too, but we will talk about that later. When we include cmath, for example, the prototypes for sqrt, pow, tan, sin, cos are all included (along with plenty of others). But we are going to write our own today instead.

3. The prototype tells us the signature of a function. If we wanted to write the a function to add two numbers together and tell us the result, what would its signature look like? Remember functions have at least three parts in their signature: return types, names and parameters.

4. Right, but today we're going to take that a step further. Here's example code of an addition function. Let's walk through what happens in the code and the order that it happens in:

```
#include<iostream>
using namespace std;

//function prototypes go here
double addition(double a, double b) {
   double my_result = a + b;
   return my_result;
}

int main() {
   double ui1, ui2, ui3; //these are for user input
   cout << addition(ui1, ui2) << endl;
   cout << addition(ui1, ui3+3) << endl;
   //what values do ui1, ui2 and ui3 have now?
}
```

5. Okay, so, now let's hit the boards. Let's design a function that will multiply two numbers together:

6. Let's write a function that takes in a string and puts "Dr." at the beginning of it. Let's call this function doctorify. An example usage might look like:

```
string my_name = "Rams";
string my_new_name = doctorify(my_name);
cout << "My new name is: " << my_new_name << endl;
```

7. Can we find the area of a triangle giving the base and height using a function?

8. When we've done this, let's write a function to compute the $\Delta V$ from the last test that we keep talking about. $\Delta V = v_e * ln(m_o/m_f)$. Remember that we use `double log(double);` from cmath to compute the natural log signified by $ln$ in mathematics. We will assume all parameters can be real numbers (doubles).

9. Okay, can we write a function to compute the hypotenuse of a right triangle given the other two side lengths? We remember that $a^2 + b^2 = c^2$ As an interesting note, the proof for this theorem was handled geometrically. It is VERY cool. Please, take a look at it or ask about it.

10. Can we write a function to round numbers to the nearest 0.5 or whole integer? For example, 3.8 would round to 4.0. 3.4 would round to 3.5. Don't forget about negatives. What would it take in? What would it output? This is directly useful in a number of applications where human readable output is desired but there is limited display capabilities (in those cases we might round to nearest 0.1 instead, but this gets us at the idea in an interesting way).