Homework #3 - Handling Strings

Name:

1 String I/O

In Python, working with strings was easy! We never had to worry about how much space a string took up or what we would do if there were spaces in it. In C++ we do have to worry about these things. When we want to handle strings, we have to create a variable with the **string** type. Here's an example:

```
1
#include <iostream>
3 #include <string>

5 using namespace std;

7 int main() {
      string name;
9
      cout << "Enter your name: ";
11    cin >> name;
      cout << "Hello " << name << endl;
13
      return 0;
15 }
```

stringIO_1.cpp

This will work fine for single-word inputs like "steve" or "jeremy" but would fail on "jeremy pedersen" or "i am a hippo" because "cin" treats words with spaces separating them as separate strings. To deal with this, C++ provides us a function called "getline()". It works a lot more like Python's raw_input():

```
2 #include <iostream>
#include <string>
4
using namespace std;
6
int main() {
8     string words;

10    cout << "Enter a sentence: ";
      getline(cin, words);
12    cout << "You said: " << words << endl;

14    return 0;
}
```

stringIO_2.cpp

2 Functions

Just like Python, C++ has a way for us to make and use functions. Functions in C++ look like this (pay attention to the top part of the program):

```
1
#include <iostream>
3
using namespace std;
5
int add(int a, int b) {
7     return a + b;
}
9
int main() {
11    int a, b;

13    cout << "Enter 2 numbers please: ";
      cin >> a >> b;
15
      cout << "The total is: " << add(a,b) << endl;
17    return 0;
}
```

function.cpp

Instead of using "def" we start our functions by saying what **type** of data they will return. And instead of using tabs to say what belongs inside the function, we use the braces "{" and "}". Otherwise, using functions is very much like you are used to from Python.

3 Now You Try

Multi-word Strings

Write a short C++ program which can ask for a name and print it back out. It should work for names that are **more than one word long** (i.e. names that have spaces in them such as "Kevin Gu" or "Jeremy Pedersen"). It should look like this when it runs:

```
Enter a name: Jeremy Pedersen
Hello, Jeremy Pedersen
```

Inputting numbers

Write a program which can change temperatures from C to F. For instance, if you type in 100, the program should give you the answer 212 (because 100 C = 212 F). You can use the formula $F = \frac{9}{5} \cdot C + 32$ to do the conversion from C to F. You should put the code to change temperature **inside a function** like this one:

```
float tempConv(float c) {
2   f = // Your code here
    return f;
4 }
```