

Homework #3 - Functions and Pointers

Name:

1 Pointers

Pointers are special variables that hold addresses. So a pointer doesn't tell you what your data is, it tells you **where to find it**. In Python, we did not use pointers because we didn't need to know where our data was: we always let the Python interpreter manage that for us.

Sometimes in C++ we need to manage our own data, or even ask the computer for extra space (for instance, if we need to make an array bigger). When we do this, the computer will give us a **pointer** telling us where in computer memory we can put our data.

Pointers are also great if we need to have a function that can change its own **operands** (the data we put into the function's parentheses "()"). For instance we could write something like this:

```
1  #include <iostream>
3  #include <vector>
5  using namespace std;
7  void minmax(int* a) {
9      if (*a > 1) {
11         *a = 1;
13     }
15     if (*a < 1) {
17         *a = 0;
19     }
21 }
23
25 int main() {
27     int x;
29     cout << "Enter a number: ";
31     cin >> x;
33
35     minmax(&x);
37
39     cout << "Result is: " << x << endl;
41
43     return 0;
45 }
```

pointerFunc.cpp

Notice that the function does NOT have to **return** any numbers (i.e. give back an answer). Instead, it directly changes the data we gave to it.

2 Now You Try

2.1 Temperature Conversion, Again

Write another function to convert temperatures from C to F. The difference is that this function should NOT return any numbers. So it will look like this:

```
1 void changeTemp(float* temp) {  
    *temp = // Your code here  
3 }
```

Note the "*" which tells the computer that the variable "temp" is actually a pointer. Also pay close attention to the example code on the page above this one. They use "*" whenever they change the data pointed to by the pointer (as in *a = 0). Doing "*a" tells the computer "go get the data at the address saved in "a", and change it". This is how the function in the example code can change "a" **directly**. Compare that to our last program, where our function had to do something like this:

```
1 F = tempConv(C);
```

Notice that C is NOT changed by the tempConv function. Instead, tempConv **returns** a value which is then saved into F.

2.2 Range Limiter

There are lots of science and engineering problems where you have to make sure that a number is between two other numbers. For instance, you might have to do some math with a number x where x is only allowed to be between 0 and 100 ($0 \leq X \leq 100$).

Write a program which asks the user to enter a number. The program then checks if the number is between -10 and 10. If the number is less than -10, the program changes it to -10. If the number is more than 10, the program changes it to 10. If it is between -10 and 10, the program does not change it. Your program should look like this when it runs:

```
Enter a number: -234234  
Changed to: -10  
Enter a number: 79789  
Changed to: 10  
Enter a number: 5  
Changed to: 5
```

Your program should use a function like this to change the numbers the user enters:

```
1 void limit(int* x) {  
    // Your code here  
3 }
```