CS 161
Intro to CS I

More Functions and
Pass by Value vs. Reference
Odds and Ends

• Assignment #4 posted...
• Start early!!!

15 lines max

functions/ lines off code
or blank lines

do one task & one task only!!
Even though it works, DO NOT USE GLOBAL VARIABLES!!!

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

int num, denom; // Make global variables, not good!

void zero_check() {
    while (denom == 0) {
        cout << "Can't divide by zero!!!" << endl;
        cout << "Enter denominator: ";
        cin >> denom;
    }
}

int main () {
    cout << "Enter numerator: ";
    cin >> num;
    cout << "Enter denominator: ";
    cin >> denom;
    zero_check(); // Call function w/o args
    cout << "Division: " << num/denom << endl;
    return 0;
}
```
```cpp
#include <iostream>

using std::cout;
using std::endl;

int pwr(int, int n=1); // Example of default args

int main() {
    int base=2, expn=8;
    cout << "The power function: " << pwr(base, expn) << endl;
    cout << "The power function: " << pwr(base) << endl;
    return 0;
}

int pwr(int x, int n) {
    int num=1;
    for(int i=0; i < n; i++) {
        num *= x;
    }
    return num;
}
```

Defaults go at end. Names of variables that are not default are not needed in prototype. n is 1.
C++ Function Overloading

• Multiple functions w/ same name
• Arguments determine function
• Default Args can be done w/ overloading
• Example: pow()
More About Functions

• Do not use global variables!
• Function Headers
  – Description, Parameters, and Return Value
  – Postconditions
  • What is this?
• Preconditions
  • What is this?
C++ Pass by Value

void swap(int, int);
int main() {
    int a=5, b=10;
    swap(a, b);
    cout << “a: ” << a << “b: ” << b;
}
void swap(int x, int y) {
    int temp = x;
    x = y;
    y = temp;
}

• What if we didn’t have temp?

OSU Oregon State University
Variables vs. Pointers

• Value Semantics
  – Values stored directly
  – Copy of value is passed
    int i, j=2;
    i=j;

• Pointer Semantics
  – Address to variable is stored
  – Copy of address is passed
    int *i, j=2;
    i=&j;
void swap(int *, int *);
int main() {
    int a=5, b=10;
    swap(&a, &b);
    cout << “a:” << a << “b:” << b;
}
void swap(int *x, int *y) {
    int temp = *x;
    *x = *y;
    *y = temp;
}
C++ Pass by Reference

void swap(int &, int &);
int main() {
    int a=5, b=10;
    swap(a, b);
    cout << "a: " << a << "b: " << b;
}
v
void swap(int &x, int &y) {
    int temp = x;
    x = y;
    y = temp;
}