CS 161
Intro to CS I

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

• Assignment #4 design due Sunday, 11:59pm
  – Submit into Assignment on Canvas!!!
• Demo Assignment 3
• Pick up Exam 1 in KEC 1174
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;
}
void swap(int &x, int &y) {
    int temp = x;
    x = y;
    y = temp;
}
Pointer and References Cheat Sheet

- *
  - If used in a declaration (which includes function parameters), it creates the pointer.
    - Ex. int *p; //p will hold an address to where an int is stored
  - If used outside a declaration, it dereferences the pointer
    - Ex. *p = 3; //goes to the address stored in p and stores a value
    - Ex. cout << *p; //goes to the address stored in p and fetches the value

- &
  - If used in a declaration (which includes function parameters), it creates and initializes the reference.
    - Ex. void fun(int &p); //p will refer to an argument that is an int by implicitly using *p (dereference) for p
    - Ex. int &p=a; //p will refer to an int, a, by implicitly using *p for p
  - If used outside a declaration, it means “address of”
    - Ex. p=&a; //fetches the address of a (only used as rvalue!!!) and store the address in p.
Demo...
Group Exercise...

Write two functions (one using references and one using pointers) that gets two C++ strings of characters as input from the user.
C++ Pass Address Implicitly
C/C++ Pass Address Explicitly