

CS 161

Intro to CS I

More Functions



Odds and Ends

- Assignment 3 demo this week
- Study sessions back to normal

② A & B | | C | | D
④

③

①

↑ precedence

~~2~~



More About Functions

- Do not use global variables!
- Function Headers
 - Description, Parameters, and Return Value
 - Preconditions
 - What is this?
 - Postconditions
 - What is this?

no declared outside fun

parameters

*pointers
rets*

Default Args



```
access.engr.orst.edu - PuTTY
1 #include <iostream>
2
3 using std::cout;
4 using std::endl;
5
6 int pwr(int, int n=1); //Example of default args
7
8 int main() {
9     int base=2, expn=8;
10
11     cout << "The power function: " << pwr(base; expn) << endl;
12     cout << "The power function: " << pwr(base) << endl;
13
14     return 0;
15 }
16
17 int pwr(int x, int n) {
18     int num=1;
19
20     for(int i=0; i < n; i++) {
21         num*=x;
22     }
23
24     return num;
25 }
"test.cpp" 25L, 388C written                               1,19   All
```

end

int x = base

n = 1

[



C++ Function Overloading

- Multiple functions w/ same name
- Arguments determine function
- Default Args can be done w/ overloading
- Example: pow()
 - <http://www.cplusplus.com/reference/cmath/pow/?kw=pow>

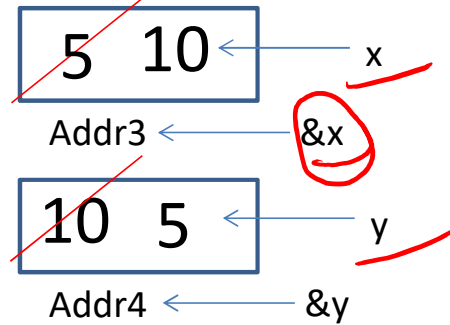
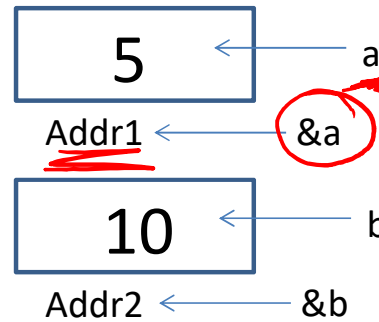
in the same scope

not return type

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;  
}
```



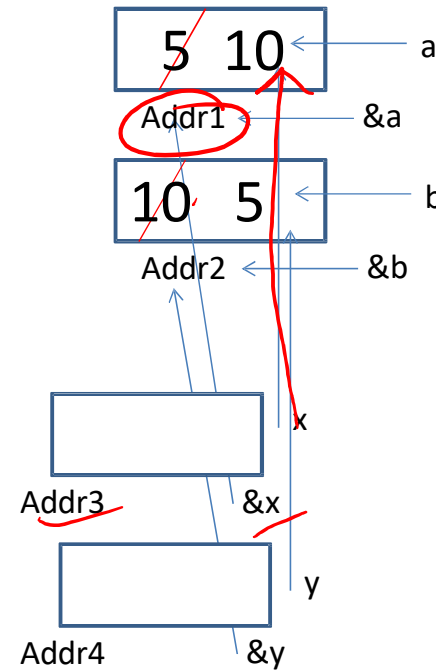
C++ Pass by Reference



```
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;  
}
```

refers to

temp



*int a=10;
int &b=a;
b=5;*

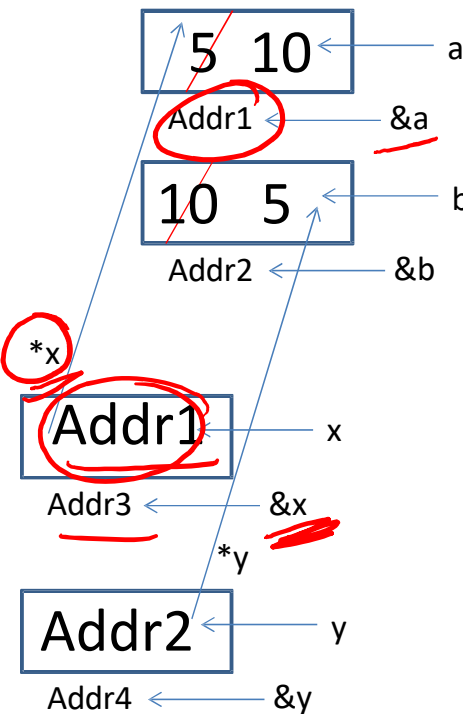
C/C++ Pointers

— more powerful



```
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;  
}
```

hold an address



```
int a=10;  
int *x;  
x = &a;  
cout << *x;  
*x = 5;
```