CS 161
Intro to CS I

Functions: Pass by Reference, Overloading, and Default Args
Chap. 4.1-4.2
Scope (Visibility)

• Part of program in which a declaration is valid
• Local variable
  – Declared inside a function only accessible inside function
• Localizing variables
  – Declaring variable in innermost scope
Illegal access outside loops

for(x = 0; x < 10; x++) {
    int y = 10;
    cout << “The value of x * y is: ” << x*y << endl;
}
cout << “The value of y is: ” << y << endl; /*y outside scope*/

• How do we fix this?
• What about if/else blocks?
Illegal access in functions

```c
int main () {
    int x=2, y=3;
    compute_sum();
    sum = x+y;  //error: sum hasn’t been declared
    return 0;
}
void compute_sum() {
    int sum = x+y;  //error: x and y outside scope
}
```
```cpp
#include <iostream>

int main() {
    // Don't declare x here if you want to use it outside of for
    for(int x=0; x++ ) {  
        int y; // Don't declare here if you want to use outside loop!!!
        std::cout << x << std::endl;
    }
    std::cout << x << "," << y << std::endl;  // scope error on x and y
    if(-1) {
        int i; // Same rule applies for if blocks
    }
    std::cout << i << std::endl;  // scope error
    return 0;
}
```
#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;
}
C++ Function Overloading

• Multiple functions w/ same name
• Arguments determine function
• Default Args can be done w/ overloading
1 #include <iostream>
2 using std::cout;
3 using std::endl;
4
5 int pwr(int, int n=1); // Example of default args
6 double pwr(double, double); // Example of overloading
7
8 int main() {
9     int base=2, expn=8;
10    double dbase=2.2, dexp=8.0;
11
12    cout << "The power function: " << pwr(base, expn) << endl;
13    cout << "The power function: " << pwr(dbase, dexp) << endl;
14
15    return 0;
16 }
17
18 double pwr(double x, double n) {
19    double num=1.0;
20
21    for(int i=0; i < n; i++) {
22        num*=x;
23    }
24
25    return num;
26 }
27
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?
### C++ Pass by Reference

```cpp
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;
}
```

OSU Oregon State University
Programming Demo

```cpp
#include <iostream>

void check_denom(float &y) {
    std::cout << "Address of ref: " << &y << std::endl;
    while(y==0) {
        std::cout << "Denominator can't be zero, enter new input: ";
        std::cin >> y;
    }
}

int main() {
    float x, y;

    std::cout << "Enter the numerator: ";
    std::cin >> x;
    std::cout << "Enter the denominator: ";
    std::cin >> y;

    std::cout << "Address of y: " << &y << std::endl;
    check_denom(y);

    std::cout << "x divided by y is: " << x/y << "\n";
    return 0;
}
```

Reading/Assignments

• Work on Assignment #3!!!
• Begin reviewing for test
Quiz #4

• Get into groups of 4-5.
• What is the difference between **break**, **return**, and **exit()**?
• Discuss Assignment #3, design a solution, and provide your algorithm using pseudocode.
  • Where do you need a loop, and what kind of loop are you going to use?
  • How are you going to use decomposition?
Conditionals, Functions, Program...

• What is the difference?
  break;
  return;
  exit(0);