CS 161
Intro to CS I

Functions: Scope & Pass by Value
Chap. 3.2 – 4.1

Odds and Ends...

• Curly brace placement and indentation
  if(x!=0) {
    printf("%d\n", x);
  } else if(x==0) {
    printf("%d\n", x);  }

• How’s Assignment #3 going???
• Test 1 Reschedule: Friday, 10/26

Scope

• 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

```c
for(x = 0; x < 10; x++) {
    int y = 10;
    printf("The value of x * y is: %d\n", x*y);
}
printf("The value of y is: %d\n", y); /*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
}
```

Parameters

- How does a function access outside information?
  - Declare global variables
  - Why is this not a good solution?

```c
int x=2;
int y=3;
int main () {
    compute_sum();
    return 0;
}
void compute_sum() {
    int sum = x+y;
}
```
Parameters: Pass by Value

```c
void compute_sum(int, int);
int main () {
    int x=2, y=3;
    compute_sum(x, y);
    return 0;
}
void compute_sum(int s, int t) {
    int sum = s+t;
    s++;  
    t++;  
    sum=s+t;
    printf("Sum is: %d\n", sum);
}
```

Parameters cont.

```c
int main () {
    int x=2, y=3;
    compute_sum(x, y);
    return 0;
}
void compute_sum(int s, int t) {
    int sum = s+t;
    s++;  
    t++;  
    sum=s+t;
    printf("Sum is: %d\n", sum);
}
• Could s and t be named x and y?
```

Common Mistakes

```c
void compute_sum(s, t);  
int main () {
    int x=2, y=3; 
    compute_sum(x, y);  
    return 0;
}
void compute_sum(s, t) {
    int sum = s+t;
    s++;  
    t++;  
    sum=s+t;
    printf("Sum is: %d\n", sum);
}
```
What if you need the sum value outside the function?

```c
int main () {
    int x=2, y=3;
    compute_sum(x, y);
    printf("Sum is: %d\n", sum);
    return 0;
}
void compute_sum(int s, int t) {
    int sum = s+t;
    s++;
    t++;
    sum=s+t;
}
```

How do we get this value?

Returning Values

```c
int main () {
    int x=2, y=3, sum;
    sum = compute_sum(x, y);
    printf("Sum is: %d\n", sum);
    return 0;
}
int compute_sum(int s, int t) {
    int sum = s+t;
    return sum;
}
```

Why are ALL these needed?

Common Mistakes

```c
int main () {
    int x=2, y=3, sum=0;
    compute_sum(x, y);
    printf("Sum is: %d\n", sum);
    return 0;
}
void compute_sum(int s, int t) {
    int sum = s+t;
    sum=s+t;
    return sum;
    sum++;
}
```

Forget to capture value

Forget return type

Forget to return value

Statements after return