

## Functions Part 3

10/27/04

CS150 Introduction to Computer Science 1

1

## Last Time

- We
  - Completed discussing reading from and writing to files
- Today we will
  - Begin learning about functions and modularity in C++

10/27/04

CS150 Introduction to Computer Science 1

2

## Program

- Write a program that calculates the area of a circle
- Write a program that calculates the area of a circle using three functions, getRadius, calcArea and outputArea

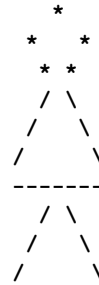
10/27/04

CS150 Introduction to Computer Science 1

3

## Program

- Write a program that draws the following stick figure using functions and top down design.



10/27/04

CS150 Introduction to Computer Science 1

4

## Program

- Write a function that takes as input two numbers and returns the larger
- Write a function that computes the average given a sum and number of elements

10/27/04

CS150 Introduction to Computer Science 1

5

## Example

```
void larger(int, int, int&);
int main()
{
    int num1, num2, large;
    cin >> num1 >> num2;
    larger(num1, num2, large);
    cout << "Larger number is " << large << endl;
}
void larger(int num1, int num2, int & large)
{
    if (num1 > num2)
        large = num1;
    else
        large = num2;
}
```

10/27/04

CS150 Introduction to Computer Science 1

6

## Passing Arguments

- Pass by value
  - Values are passed into the function
  - Any changes made in the function are not reflected in the main
- Pass by reference
  - Any changes made in the function are reflected in the main

10/27/04

CS150 Introduction to Computer Science 1

7

## Example

```
void swap(int &, int &);
int main(void)
{
    int i, j;
    cin >> i >> j;
    swap(i,j);
    cout << i << j;
    return 0;
}

void swap(int & num1, int & num2)
{
    int temp;
    temp = num1;
    num1 = num2;
    num2 = temp;
}
```

10/27/04

CS150 Introduction to Computer Science 1

8

## Rules for Parameter Lists

- There must be the same number of actual and formal arguments
- The correspondence between actual and formal arguments is by position only
- Corresponding actual and formal arguments must match in type
- The names of the actual and formal arguments may be the same or different
- For reference arguments only, the actual argument must be a single, simple variable

10/27/04

CS150 Introduction to Computer Science 1

9

## Example

- Given the following function prototype:  
`void checkIt(float &, float &, int, int, char &);`
- And declarations in main:  
`float x, y;`  
`int m;`  
`char next;`
- Which are legal?  
`checkIt(x, y, m+3, 10, next);`  
`checkIt(m, x, 30, 10, 'c');`  
`checkIt(x, y, m, 10);`  
`checkIt(35.0, y, m, 12, next);`  
`checkIt(x, y, m, m, c);`

10/27/04

CS150 Introduction to Computer Science 1

10

## What is the output?

```
void changeIt (int, int&, int&);
void main()
{
    int i,j,k,l;
    i = 2;
    j = 3;
    k = 4;
    l = 5;
    changeIt(i,j,k);
    cout << i << j << k << endl;
    changeIt(k,l,i);
    cout << i << k << l << endl;
    changeIt(i,j,j);
    cout << i << j << endl;
    changeIt(i,i,i);
    cout << i << endl;
}

void changeIt(int a, int& b,
int& c)
{
    a++;
    b += 2;
    c += a;
}
```

10/27/04

CS150 Introduction to Computer Science 1

11

## What is the output?

```
void changeit(int, int&, int&);
void main()
{
    int i,j,k,l;
    i = 2;
    j = 3;
    k = 4;
    l = 5;
    changeit(i, j, k);
    cout << i << j << k << endl;
    changeit(k,l,i);
    cout << i << k << l << endl;
}

void changeit(int j, int&
i, int& l)
{
    i++;
    j += 2;
    l += i;
}
```

10/27/04

CS150 Introduction to Computer Science 1

12

## Program

---

- Write a function to compute the sum and average of two integers, and return the values of sum and average.
- An example function call would look like:
  - `compute (4, 5, sum, average);`

10/27/04

CS150 Introduction to Computer Science 1

13

## Summary

---

- In today's lecture we covered
  - Library functions
  - Programmer defined functions
- Readings
  - P. 170 - 180

10/27/04

CS150 Introduction to Computer Science 1

14