

# **16.216: ECE Application Programming**

Summer 2013

## Lecture 5: Key Questions July 25, 2013

1. Describe the basic use and syntax of functions, including return types and arguments.

2. What is a function prototype? When and why is it necessary?

3. Explain the idea of scope and how it relates to functions. Also, explain what happens when function arguments are passed by value.

You may wish to refer to the following example:

```
#include <stdio.h>
#include <math.h>

double hyp(double a, double b);

void main()
{
    double x,y,h;
    printf("Enter two legs of triangle: ");
    scanf("%lf %lf",&x,&y);
    h=hyp(x,y);
    printf("Trgle w legs %lf and %lf has hyp of %lf\n",
           x,y,h);
}

double hyp(double a, double b)
{
    double sum, result;
    sum = a*a + b*b;
    result = sqrt(sum);
    return result;
}
```

4. **Example:** What does the following program print?

```
#include <stdio.h>

int f(int a, int b);           // Function prototype

int main() {
    int x = 1;
    int y = 2;
    int result1, result2, result3;

    result1 = f(x, y);
    result2 = f(y, result1);
    result3 = f(result1, result2);

    printf("x = %d, y = %d\n", x, y);
    printf("Result 1: %d\n", result1);
    printf("Result 2: %d\n", result2);
    printf("Result 3: %d\n", result3);

    return 0;
}

int f(int a, int b)           // Function definition
{
    int i;                    // Loop index
    int r = 0;                // Result

    for (i = 0; i < a; i++)
        r += b;

    return r;
}
```

5. **Example:** Write a function to do each of the following:
  - a. Prints a series of `LINE_LENGTH` dashes on a single line, where `LINE_LENGTH` is a predefined constant (using `#define`)
  - b. Reads a value from the console input and returns 1 if the value is even, 0 if it's odd
  - c. Takes four numbers as arguments and returns their average

6. Explain what a pointer is, and how we can use them in C.

7. Explain the use of passing function arguments by address.

8. What does the following program print?

```
#include <stdio.h>
#include <math.h>
void get_r_theta(double a, double b,
                 double *adr_r, double *adr_th);

void main()
{
    double x,y,h,r,th;
    printf("Enter x, y components of vector: ");
    scanf("%lf %lf",&x,&y);
    get_r_theta(x,y,&r,&th);
    printf("Vector with x=%lf and y=%lf
           has r=%lf, theta=%lf\n",x,y,r,th);
}

void get_r_theta(double a, double b,
                 double *adr_r, double *adr_th) {
    double sum;
    sum = pow(a,2)+pow(b,2); //or a*a+b*b;
    *adr_r = sqrt(sum);
    *adr_th = atan2(y,x);
}
```

9. **Example:** What does the following print?

```
int f(int *a, int *b);

int main() {
    int x = 1;
    int y = 2;
    int result1, result2, result3;
    result1 = f(&x, &y);
    result2 = f(&y, &result1);
    result3 = f(&result1, &result2);
    printf("x = %d, y = %d\n", x, y);
    printf("Result 1: %d\n", result1);
    printf("Result 2: %d\n", result2);
    printf("Result 3: %d\n", result3);
    return 0;
}

int f(int *a, int *b)
{
    int copyB = *b;
    while (*a > 1) {
        *b += copyB;
        (*a)--;
    }
    return *b;
}
```

10. Write a function that:

- Given two integer arguments,  $x$  and  $y$ , store the quotient and remainder of  $x / y$  into locations specified by arguments  $q$  and  $r$ , respectively.

- Uses pointers to swap the values of two double-precision variables



PE3: Functions

This exercise functions on the “change problem”—write a program that will, given an amount of change to be returned, determine the minimum number of coins required to fulfill that amount. In this specific problem, the amount is \$2.00 or less, and the coins available are half dollars, quarters, dimes, nickels, and pennies.

Use the space below and on the following page to draw a flowchart for this program.

Under what circumstances do you use functions? Identify the best opportunity to use a function in this program, and draw a flowchart that incorporates the function, as well as calls to that function.