

IPC144 Lab #5 - Functions (Mr. Saul's Sections)

Weekly Tip: The value of each argument in a function call is copied into the corresponding function parameter. The scope of each function parameter is the body of the function. Each parameter is distinct from its corresponding argument in the function call.

In this lab, you will work with functions. As with the previous labs, if you get stuck, don't be shy about asking for help - that is what the lab sessions are for.

NOTE: When you have completed this lab, and placed answers into the file called "lab5.txt", follow the instructions (at the bottom of this lab) to submit your answers to Murray Saul's e-mail account...

Start by compiling and testing the following program (call it lab5a.c):

```
#include <stdio.h>
#include <math.h>
/* The distance between two points is the square root of
the sum of the squares of the differences between the
matching coordinates.
This function returns the distance between points (x,y)
and (x1,y1). The return value is a double.
*/
double distance(double x, double y, double x1, double y1) {
    double dx, dy;
    dx = x - x1;
    dy = y - y1;
    return sqrt(dx * dx + dy * dy);
}
main() {
    double x1, y1, x2, y2;
    printf("Enter the x and y coordinates of a point: ");
    scanf("%lf %lf", &x1, &y1);
    printf("Enter the x and y coordinates of another point: ");
    scanf("%lf %lf", &x2, &y2);
    printf("The distance between those two points is %.3lf\n",
        distance(x1, y1, x2, y2));
}
```

1. On phobos, no matter how correctly you've typed in this program, it won't compile properly even though no syntax errors are found. You should receive an error message to the effect that the compiler cannot find the sqrt (square root) function, which is declared in the header file <math.h>. This is because on Unix systems, the "math" library is not normally searched, unless you specifically indicate that it should be searched. In Unix, the library whose functions are declared in <math.h> is /usr/lib/libm.a You can specify this library by compiling as follows:

```
cc lab5a.c /usr/lib/libm.a
```

Test the program by using "0 4" and "3 0" as the data points.

There is a shortcut that doesn't require you to remember the library name and path. You can search the math library by adding the switch "-lm" to the compile command line. The command line switch -lm means "also search the library named libm.a in the standard library directory". For example, you can recompile with the command:

```
cc lab5a.c -lm
```

Note that the library whose functions are declared in <stdio.h> is automatically searched, so you don't have to state that this particular library should be searched.

2. Carefully note the relationship between the main function's variables x1, y1, x2, and y2, and the distance function's variables x, y, x1, and y1.

To which variables in distance are main's x1 and y1 related?

To which variables in distance are main's x2 and y2 related?

Note that the dx and dy variables used in distance are totally unknown to main. We say that the scope of these variables is the body of the function distance.

3. Copy lab5a.c to lab5b.c. In lab5b.c, change the order of the "distance" function and the "main" function so that main appears before distance.

(If you are using nled to type in your programs, an easy way to do this is to move to the start of main and press <Esc> then **b** to "mark the beginning of a block". Then move just past the end of main and press <Esc> then **e** to "mark the end of the block". Finally move just before the distance function and press <Esc> then **m** to "move the block".)

Now recompile your reordered source. What message did you get?

Now declare the distance function before main, by prototyping it with the line:

```
double distance(double x, double y, double x1, double y1);
```

Note the semi-colon at the end of this statement, as compared to the function heading where no semi-colon follows the parentheses.

Now recompile your updated source. Did it successfully compile?

In general, if you don't want to worry about the order in which you write functions, you should prototype each function before you start to code any of them. Note that you never need to prototype main, however, since your own code will never call it.

4. Copy lab5b.c to lab5c.c. In lab5c.c, take out the names x, y, x1 and y1 from the distance function prototype (but not from the function header line for distance). Now recompile your source code. Did it compile successfully? _____

Note that the names that you give to the parameters are only important when you code the function distance. If you are specifying a function prototype the compiler only needs to know the datatype of each parameter; names are optional for parameters in function prototypes.

5. Copy lab5c.c to lab5d.c. In lab5d.c, change the data types of the main function's x1, y1, x2 and y2 variables from double to int (don't forget to also change the format strings in the scanf statements), but leave the distance function unchanged. Recompile your source and test the program. Did you get any error messages? _____

One of the purposes of prototyping a function before calling it is to let the compiler know if it needs to perform any automatic data conversions (int to double, in this case) when copying the argument data to the function.

6. [If you have time] Modify the program so that it takes 3 dimensional coordinates (x, y and z coordinates) for each point. Note: The distance between two 3-D points is the square root of the sum of the 3 squares of the differences between all three matching coordinates.

Submission Requirements:

If you are in Murray Saul's class, issue the following command to send your lab #5 answers to Murray Saul:

```
mail -s "144lab5" -c $USER@learn.senecac.on.ca murray.saul@senecac.on.ca <lab5.txt
```

The option **-s "144lab5"** makes subject line appear as "144lab5" so instructor can filter these e-mails in a directory to collect all lab5 submissions.

The option **-c \$USER@learn.senecac.on.ca** sends a copy of the e-mail message to YOUR learn account. The variable **\$USER** is your Matrix id name assuming that you are issuing this command when logged into your Matrix account. Please keep this e-mail for the remainder of this term as proof that you sent your lab by the required deadline...