

An Introduction to C Programming

Ryan Baker

October 1, 2014

What is C?

- It's an older programming language, C++ is the modern equivalent.
- C is essentially comprised of functions, and in these functions you can use variables, conditional statements, and loops to store and manipulate data.
- Used by physicists to analyze data, run simulations, etc.

Using the Terminal

- Mac uses a UNIX terminal (same as Linux).
- Allows you to navigate directories and use powerful console commands.
- **pwd** tells you the current directory.
- **ls** lists everything in the current directory.
- **cd example** changes the directory to example.
- **cd ..** moves one directory up.
- **mkdir folder** makes a new directory named folder.
- **rmdir folder** removes the directory named folder.
- **cp file1 file2** copies file1, naming it file2.
- **mv file1 file2** moves file1 to file2.
- **rm file1** removes file1 (be careful, permanent!)

Writing Code

- Code is written in a text editor (similar to notepad)
- **vi** is used often here.
- **vi filename.c** opens filename.c in vi.

Useful vi Commands:

- **i** enters insert mode, allowing you to type
- **escape** leaves insert mode
- **:w** writes (saves) the file
- **:q** quits out of vi
- **:q!** quits vi without saving the file

Compiling Code

- Once your program is written it must be compiled before it can be run.
- Here we use the GNU C Compiler, **gcc**
- **gcc filename.c -o example -lm**
 - This compiles "filename.c", into an executable program which is run by typing "./example" in the terminal.
 - The "-lm" is needed only when the program uses math functions, such as "sqrt".

Simple Example Code

- This code simply writes "Hello World!" on the screen when it is run.
- **#include** loads a library of functions before the program is run.
- **int main()** tells the compiler that the program is running the function called "main" and that it will return an integer when it is finished.

```
#include <stdio.h>

int main()
{
    printf("Hello World! \n");
    return 0;
}
```

- **printf** prints "Hello World!" to the screen. The `\n` moves the cursor to the next line.

Different Types of Variables

- When you create a variable or a function you must assign it a variable type.
- **float** holds a number with a decimal point.
- **char** holds a string of characters.
- **int** holds an integer value (no decimal place).

```
float n = 1.5;  
char name;  
name = "Ryan";  
int a, b, c, d;
```

Manipulating Variables

- Line 1: Sets the variable a equal to 12.
- Line 2: Sets b equal to its old value plus 3.
- Line 3: Checks to see if c is equal to 10.
- Line 4: Checks to see if d is less than 10.

```
a = 3*4;  
b = b + 3;  
c == 10;  
d < 10;
```

If Statements

- If statements allow you to section off a part of your code, only running it if a conditional statement is true.
- Here printf is never called due to the if statement.

```
#include <stdio.h>

void main()
{
    if(3 > 5){
        printf("3 is greater than 5 \n");
    }
}
```

For Loops

- A loop allows you to run the same section of code more than once without rewriting everything out.
- A for loop will continue running until the conditions set in the for loop have been met.
- Three conditions are set in a for loop: initializing the variable, the condition to be met, and the way the variable is updated at the end of a loop.

For Loop Example

- In this program the value of `x` is printed to the screen each time the loop runs.
- `x` is set to 0, and the loop continues as long as `x` remains less than 10.
- Each time the loop resets 1 is added to the value of `x` (`x++` is the same as `x = x + 1`).

```
#include <stdio.h>

void main()
{
    int x;
    for (x = 0; x < 10; x++){
        printf("%d \n", x);
    }
}
```

More Complex Example

```
#include <stdio.h>

int add ( int x, int y)
{
    return x + y;
}

void main()
{
    int x, y;
    printf("Enter two numbers to add: ");
    scanf("%d", &x);
    scanf("%d", &y);
    printf("The sum of your numbers is %d\n", add (x,y));
}
```