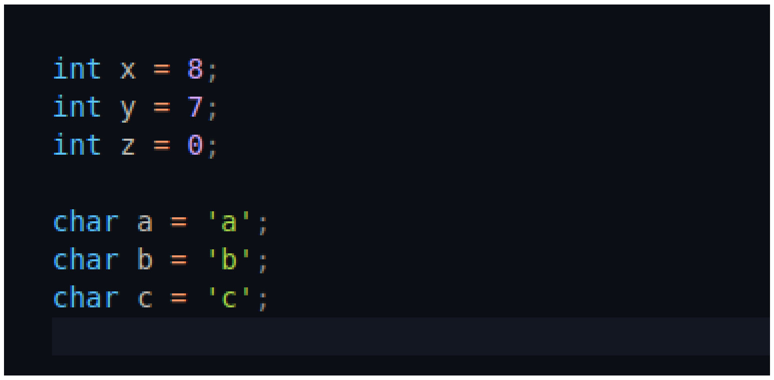
Lecture 8: Data types & “main” function

C as a Static language:

As we have seen, C is a static language, which means that we will need to explicitly tell the compiler what are the types of the data that we are going to be using in our code, and to do so, we will be going with the following way of declaring our variables to use them afterwards in our code:

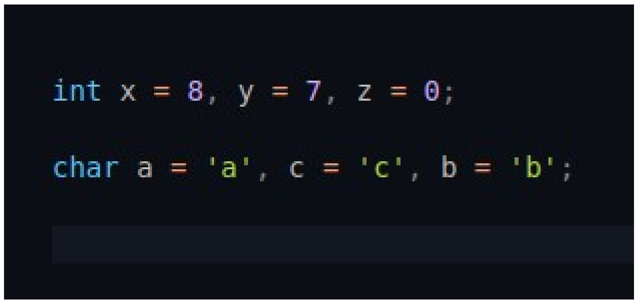
1. Declaring each variable on its own line:



```c
int x = 8;
int y = 7;
int z = 0;

char a = 'a';
char b = 'b';
char c = 'c';
```

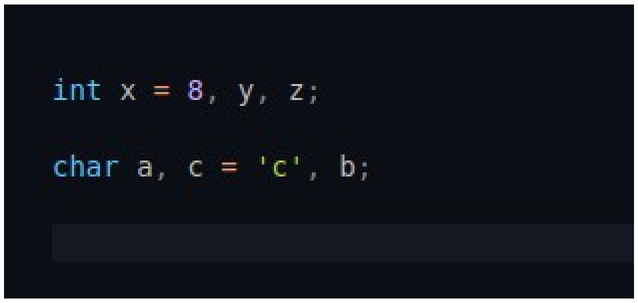
2. Declaring similar types of variables on the same line:



```c
int x = 8, y = 7, z = 0;

char a = 'a', c = 'c', b = 'b';
```

3. Either way of declaration is right to go with, and since C is a weakly typed language, we can even declare the variable without giving it any value when we declare it:



```c
int x = 8, y, z;

char a, c = 'c', b;
```

Data types in C:

The basic data types that we will be looking at in this lecture are gong to be the following:

1. `char`

This is the data type that is responsible of storing the characters that we find on the keyboard that represent the characters that we use in different languages. It is stored in the memory in 1 byte cells and is converted by the C compiler to its equivalent ASCII code before it gets stored.

What is ASCII code?

ASCII (American Standard Code for Information Interchange) is a character encoding standard that maps characters to numeric values. It uses **7 bits** to represent 128 characters, including:

- Printable characters (`A-Z`, `a-z`, `0-9`, punctuation marks, etc.)
- Control characters (e.g., newline, tab)

Examples of ASCII Mappings:

- `'A'` → `65`
- `'a'` → `97` `'0'` → `48`
- Space (`' '`) → `32`

So the char variables are first converted to the ASCII representation first, and then the representation is converted into bits to be stored in the memory.

2. `int`

Integers are the type that we use to deal with whole numbers that has no decimal point. They take 4 bytes in the memory.

3. `float`

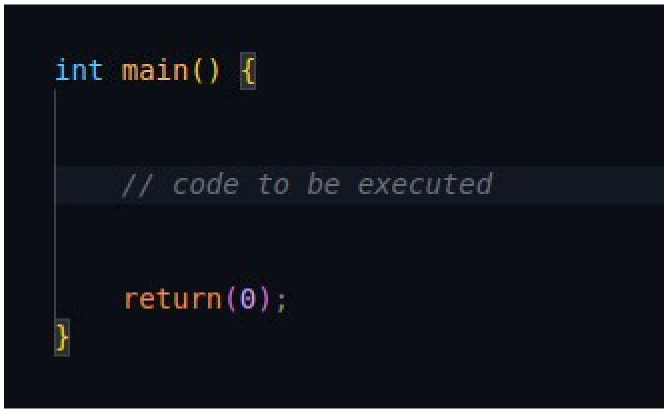
Floating point numbers on the other hand are what we use to deal with numbers that has decimal point. They also take 4 bytes in the memory.

Check out other data types to see what are other types that we also deal with in C.

“main” function:

As we know, C is a procedural language, which means its first concern is to encapsulate procedures to and use them as we go through our project.

The fundamental procedure of any code we write is going to be the main( ) function that carries on the execution of any code starting from the first line in it.



```c
int main() {

    // code to be executed


    return(0);
}
```

We usually end the main( ) with a “return(0);” statement, but why?

Each operating system when it starts any program of service, it sees it as a process that it needs to monitor its operation to make sure there were no errors in the execution of it that it needs to notify us about. The SUCCESS or FAILURE of a process is determined by what we call an “exit code”.

This exit code is a value that we pass at the end of any process or “procedure” to indicate the state of the execution of the process itself. The standard value to passed at SUCCESS is set to be 0, and the standard value to be passed at FAILURE is set to be 1.

Therefore, we return a 0 value at the end of our main( ), or in other words, the end of our process that is carried out in the man( ), which is the beginning of the execution of our code, and the procedure that encapsulates the whole code, to indicate the success of our code.

And to be able to pass a value of 0, we will need to define this value in the memory as “int”, that is the reason we define the data type on the main( ) as int in its declaration.