Here are some aspects of C that you will need to learn for the first project.

- basic types we will use: int, char, float and their variants (e.g. you can have an “unsigned char” or an “unsigned int”). A char is a 1-byte integer type, generally used to store the integer code for a character (e.g. 65 for A).

- You can use decimal notation for number (e.g. 1.0 or 1) or hexadecimal notation (e.g. 0x010D).

- arithmetic operators: +, -, *, /, % (modulus/remainder)

- ! means “not”

- The basic memory layout is a linear chunk of memory with the stack growing from the top, and the heap at the bottom. All memory must be explicitly allocated. Either statically (e.g. int a) on the stack, or dynamically (e.g. using malloc) on the heap. If memory is allocated with malloc it should be deallocated with free.

- When a variable is declared, memory is created for it on the stack. If you intend to use the variable to point to something on the heap, you declare it as a pointer type, and the value stored on the stack is an address of memory on the heap.

- A pointer is an address of some other area in memory (either the stack or the heap). The simplest way to understand pointers is to play around with a scalar value on the stack. Suppose we have.
float f = 3.0;

- To declare something as a pointer, use *, e.g.
  
  float *fptr; // fptr is a pointer to one or more floats; // (or rather, will be, once the space has been allocated)

- To make the pointer fptr point to f, we need to provide it the address of f. To do that, we use the “address of” operator, i.e.
  
  fptr = &f;

- To dereference a pointer, i.e. to “follow” the pointer to see what value is at the end of it, we use the “contents of” operator, which is again an asterisk.
  
  printf("A points to a float with value %f\n", *fptr);

- The constant NULL is used to indicated a null pointer. So, if you want to initialize a pointer with a value to indicate it hasn’t been set yet, use NULL.

**Example: Swapping**

We write a function to swap the values of two char variables.

```c
void swap(char *a, char *b) {
    char tmp = *a;
    *a = *b;
    *b = tmp;
}
```

And here is an example of code that calls it

```c
char c1 = 4;
char c2 = 5;
swap(&c1, &c2);
```