# Module 1 – Numbers, Variables, Arithmetic Operations, and Math Expressions in C

Alternate Reference: WikiBooks.org [C Programming - Variables](https://en.wikibooks.org/wiki/C_Programming/Variables)

[Variables and Math](http://learn.parallax.com/propeller-c-start-simple/variables-and-math)9

This first week’s work hints at the limitations of the machine when dealing with numbers and applying the math operations. Memory space is limited in size and access (“bus width”) forcing repetitive and “reiterative” operations to do a simple multiplication. The arithmetic engine that performs the math operations has these same limitations.

Therefore, in programming in general, the number size and type you require must be declared ahead of time before any math operations or even references to that number can happen. The section on variables shows you how this is done.

A more formal discussion is cited above. These first few weeks of work are critical foundation. Use that discussion room!

Insert a picture here and paste in a copy of your program for **Try This** for **Variables and Math**. Don’t forget to adjust the comments for the program.

Insert a picture here and paste a copy of your program for **Your Turn** section of **Variables and Math**. Comment the program!

## Module 1 – [Floating Point Math](http://learn.parallax.com/propeller-c-start-simple/floating-point-math)

Floating point math is a special section in the C library and deserves special treatment because they are special numbers. Some microcontrollers and microprocessors have special circuits (Coprocessors) to deal with this variable type.

Insert a picture here and paste in a copy of your program for **Try This** for **Floating Point Math**. Don’t forget to adjust the comments for the added lines in the program.

Insert a picture here and paste a copy of your program for **Your Turn** section of **Floating Point Math**. Comment the program!

## Module 1 – [Array Variables](http://learn.parallax.com/propeller-c-start-simple/array-variables)

An array gives a group of numbers context (purpose and meaning) and order. The idea of order suggests that each element (number) in the array has a “place” within that group. If that is the case, then each “place” has a number assigned to it called an address (or index). The result is that we can refer to any element in the array simply by its address without regard for the value contained there.

In addition, arrays can have multiple dimensions:

1D arrays can be referred to as ordered lists.

2D arrays can be referred to as ordered fields. This is like the screen you’re looking at right now. Each pixel on the screen in front of you can be directly addressed by the computer. You are looking at a field of pixels. Each pixel address holds a color value.

3D arrays can be referred to as ordered volumes (?), solids (?). This has recently become an important subject in the world of manufacture in 3D printing.

Insert a picture here and paste a copy of your program for **Try This** section of **Array Variables**. Comment the program!

Insert a picture here and paste a copy of your program for **Your Turn** section of **Array Variables**. Comment the program!