# Programming

In order to help you understand the code above a brief introduction to programming will be presented in this section. This is not intended to fully develop the concepts, just to help you understand the code you are using. Learning a programming language for the first time can be intimidating as learning a new spoken language, but with repeated practice you will pick it up surprisingly quickly.

• **Variables**: Variables are pieces of information which may change during different runs of the program. This is why they are called “variable”, because their value can vary. There are two main features to think about with a variable. The declaration and the initialisation.  
  
**Declaration:**  
The variable declaration will state what type of variable you intend to use as well as what you will refer to it as in your program. For this course, you will most likely be working with Integers (int) which are whole numbers, or Doubles (double) which are real numbers (i.e: decimals). You may occasionally need to use a String, which is just a set of characters, for example: “This is a String”. The declaration of a variable states what the variable will be, as well as what you will refer to it as in your program. For example:

int x;

This line if code instructs your program to reserve a place in its memory so that you can store an integer value. It has also instructed your program that if you wish to access that space in memory, you will do so by using its name “x”. Note that at this point, there is no value associated with this variable.

**Initialisation:**When a variable is assigned its first value, it can be thought of as being initialised. If you have already declared your variable, you will **not** need to re-declare it, you can simply tell the program which variable you wish to work with, and assign its value. For example:

x = 5;

This line of code instructs your program to access the memory location where x is stored, and keep a record of the value “5”.

**Doing it all at the same time**  
If you try to use your variable before it has been initialised, your program may have errors, because it cannot find a value to use. For this reason, it is sometimes considered good practice to declare and initialise your variables at the same time. For example**:**

int x = 5;

This line if code instructs your program to reserve a place in its memory, which holds a value of 5, and you can access with “x”.  
  
Note that it is also conventional to list variables at the start of your program or function. By following this convention, it is easy for you to see what variables you have in your program or function.

*But why do I have to do this??? Why can’t I just declare them whenever I want???*

*When you start working, you will often be working with other people’s programs. That is, there may come a day were you need to work on someone else’s program, or they need to work on yours. By following standard conventions, you are making it easier for yourself and others to work on different bodies of code. Always follow the conventions! You don’t want to be “that guy” who doesn’t.*

Once we have a variable it can be used in an operation to help us get a required output. For example, if I needed to know how many robots kits I needed. I would first need to know how many people are in the room so I would make a variable:

int peopleInRoom = 32;

Here, peopleInRoom is a variable of integer type. The variable is assigned value: 32. I would also want a variable for the number of people sharing each kit:

int peoplePerKit = 4;

Here, peoplePerKit is a variable of integer type. The variable is assigned value: 4. Now we can do some maths to figure out how many kits we require, the whole sequence looks like:

int peopleInRoom = 32;

int peoplePerKit = 4;

int kitsNeeded = peopleInRoom/peoplePerKit;

We just declared kitsNeeded , a variable of integer type and it holds the solution of peopleInRoom (32) / peoplePerKit (4).

* The most common variable type you will use in this project is an integer (int). An int stores a positive or negative whole number between -32768 (or (-215)) and +32767 (or 215). It is the most appropriate type to use for most applications in this project. For example, storing values read from sensors. See the Arduino reference page [https://www.Arduino.cc/en/Reference/ Int.](https://www.Arduino.cc/%20en/Reference/Int)

**Common issues to look out for when working with Integers:**

In the case above, 32/4 is easy because the result is 8. So kitsNeeded will always come out with a nice whole number. However, what if we had 3 members per group.

int peopleInRoom = 32;

int peoplePerKit = 3;

int kitsNeeded = peopleInRoom/peoplePerKit;

Naturally give us a result of 10 . However, in the code above, kitsNeeded would be listed as 10. When we divide an integer by another integer, the program will truncate the result down to the nearest whole number. This is known as integer division.