Some background regarding programming


The text below is to a large extent from: http://openbookproject.net//thinkCSpy/

Computers take many forms and are available in many places in people's lives. From special purpose computers, such as those that monitor operation in your car, through general purpose computers such as those used in homes and offices, to specialized computers constructed to solve very large problems in mathematics and science, all computers perform tasks directed by instructions (software, programs) that someone wrote. People in many walks of life can simply purchase software to assist in their work. Others, whose work is more specialized or who need specific variations in the operation of the computer, need to be able to construct software for their own purposes.

Software development, or computer programming, now takes many forms. In this project, either "higher-order languages" or "visual programming" may be used. In both of these, a piece of software exists to allow human-friendly forms of expression to be translated to the highly specific form of instructions to which computers actually respond. These translation software tools include "compilers", "interpreters", and "development environments". The purpose of all of them is to allow you to work in a manner that is at least reasonably familiar, using simple operations such as picking items from a menu list, or giving written instructions in a language that looks no worse than a mixture of English and algebra. (Unlike English, with its 2,000,000 or so different words, a computer language typically has only about 100 words. So, many find learning a programming language far easier than learning a second human language.)

In the programming section of this tutorial, you will learn to use a programming language (Scratch) to construct, modify, and analyze computer programs of your own. There are many programming languages available, but the one most used in introductory programming experimentation is still BASIC (standing for Beginner's All-purpose Symbolic Instruction Code). Although not containing many of the powerful capabilities developed by the software industry in the last 30 years, BASIC has a near-English appearance that is often appreciated by beginners. A very large number of books are available on BASIC, and on exploratory programming projects for beginners using BASIC. You will be using BASIC to command a microcontroller in the sensors section of the tutorial.

Computer programming languages (from: http://computerprogramming.suite101.com/article.cfm/programminglanguages):

Computer programming languages can be broken down into several categories - mark-up languages such as HTML allow plain text to be interpreted in special ways, depending on the exact requirements of the programming language.
*Interpreted programming languages* are similar in that they are designed to be processed by an application (such as Word or Internet Explorer), which takes plain text, easy to understand, instructions, and translates them into actions inside the target application.

The last category of computer programming languages need to be treated by a special application called a compiler, to produce a native executable. *Compiled languages* result in applications that can be run in the operating system (Windows, Linux, MacOS, PalmOS, etc.) rather than needing an interpreter.

There is a special category of programming language containing languages such as Java, in which the compiler creates a special file, known as byte code, which needs an additional application (Virtual Machine) to execute the code. This combines the flexibility of an interpreted language (write once, run anywhere that is supported by the application) with the speed of a compiled language.