

CHAPTER 2: TYPES, OPERATORS AND EXPRESSIONS

Variables and constants are the basic data objects manipulated in a program. Declarations list the variables to be used, and state what type they have and perhaps what their initial values are. Operators specify what is to be done to them. Expressions combine variables and constants to produce new values. These are the topics of this chapter.

2.1 Variable Names

Although we didn't come right out and say so, there are some restrictions on variable and symbolic constant names. Names are made up of letters and digits; the first character must be a letter. The underscore "_" counts as a letter; it is useful for improving the readability of long variable names. Upper and lower case are different; traditional C practice is to use lower case for variable names, and all upper case for symbolic constants.

Only the first eight characters of an internal name are significant, although more may be used. For external names such as function names and external variables, the number may be less than eight, because external names are used by various assemblers and loaders. Appendix A lists details. Furthermore, keywords like `if`, `else`, `int`, `float`, etc., are *reserved*: you can't use them as variable names. (They must be in lower case.)

In modern C languages, the limitation of the first eight characters of a variable name being unique has been extended. In most C variants you can expect that at least 30 characters of a variable name are treated as unique. The eight character limitation was to reflect a typical limitation of identifier length in assembly language programming and run-time linkers of the time.

Naturally it's wise to choose variable names that mean something, that are related to the purpose of the variable, and that are unlikely to get mixed up typographically.

2.2 Data Types and Sizes

There are only a few basic data types in C:

<code>char</code>	a single byte, capable of holding one character in the local character set.
<code>int</code>	an integer, typically reflecting the natural size of integers on the host machine.
<code>float</code>	single-precision floating point.
<code>double</code>	double-precision floating point.

In addition, there are a number of qualifiers which can be applied to `int`: `short`, `long`, and `unsigned`. `short` and `long` refer to different sizes of integers, `unsigned` numbers obey the laws of arithmetic modulo 2^n , where n is the number of bits in an `int`; `unsigned` numbers are always