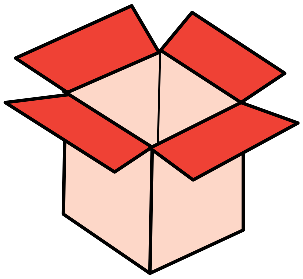
Creating program variables



A variable is a container in a C program in which a data value can be stored inside the computer's memory. The stored value can be referenced using the variable's name. The programmer can choose any name for a variable providing it adheres to the naming conventions listed in the table below:

Naming rule:	Example:
CANNOT contain any of the C keywords	**volatile**
CANNOT contain arithmetic operators	**a+b*c**
CANNOT contain punctuation characters	**%$#@!**
CANNOT contain any spaces	**no spaces**
CANNOT start with a number	**2bad**
CAN contain numbers elsewhere	**good1**
CAN contain mixed case	**UPdown**
CAN contain underscores	**is_ok**

Beware

Variable names are case-sensitive in C – so variables named "VAR", "Var", and "var" would be treated as three separate variables.

It is good practice to choose meaningful names for variables to make the program code more easily understood. To create a variable in the program simply requires it to be "declared". A variable declaration has this syntax:

data-type variable-name ;

First, the declaration specifies which type of data the variable is permitted to contain. This will be one of the four data types described on the opposite page. The data type is followed by a space then the chosen variable name, adhering to the naming conventions in the table above. As with all statements in C programs, the declaration must end with a semi-colon terminator. Multiple variables of the same data type can be created in a single declaration as a comma-separated list, like this:

data-type variable-name-1 , variable-name-2 , variable-name-3 ;

…cont'd

There are four basic data types in the C language. These are defined using the C keywords that are listed in the table below, together with a description of each data type:

Data type:	Description:	Example:
char	A single byte, capable of storing just one character	**'A'**
int	An integer whole number	**100**
float	A floating-point number, correct to six decimal places	**0.123456**
double	A floating-point number, correct to 10 decimal places	**0.0123456789**

Beware

Values of the **char** data type must always be enclosed by single quotes – double quotes are incorrect.

The four data types allocate different amounts of machine memory for storing data. The smallest is the **char** data type, which allocates just a single byte of memory; and the largest is the **double** data type, which typically allocates eight bytes of memory. This is twice the amount of memory allocated by the **float** data type so the **double** data type should only be used when a lengthy, precise floating-point number is a necessity.

Variable declarations should be made before any executable code appears in the program. When a value is assigned to a variable, that variable is said to have been "initialized". Optionally, a variable may be initialized when it is declared.

Hot tip

In a variable assignment the part to the left of the = operator is known as the L-value object and that to its right is known as the R-value data.

The example code fragment below declares and initializes various variables with appropriate values as described in the code comments – descriptive comments between **/*** and ***/** that are ignored by the compiler:

```
int num1 , num2 ;         /* Declares two integer variables. */

char letter ;             /* Declares a character variable. */

float decimal = 7.5 ;     /* Declares and initializes
                                    a floating-point variable. */

num1 = 100 ;              /* Initializes the integer variables. */
num 2 = 200 ;

letter = 'A' ;            /* Initializes the character variable. */
```