Naming constants in a program is called as **macro definition**, #define CONSTANT\_NAME “value”

If the value is like this, (5.0f/9.0f) then do not avoid append f because then C may round off the value of 5/9.

Identifiers are names of variables, functions, macros, etc. The identifiers can contain only alphabets, numbers and underscore. It cannot begin with number or underscore.

C is case sensitive therefore dog, Dog, DoG are all different variables.

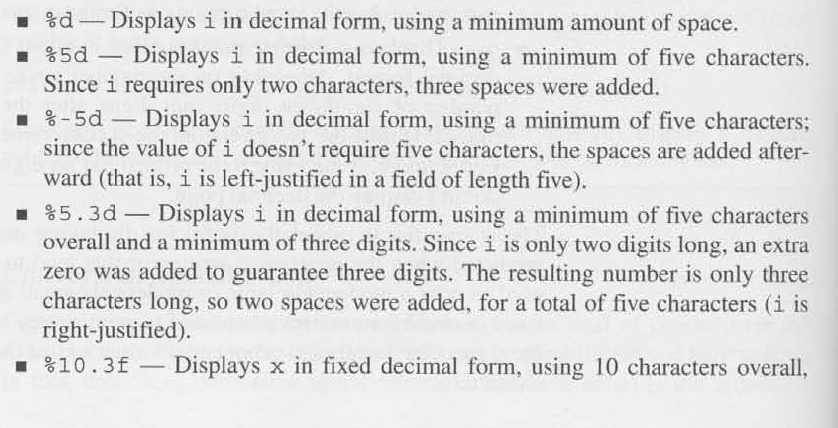
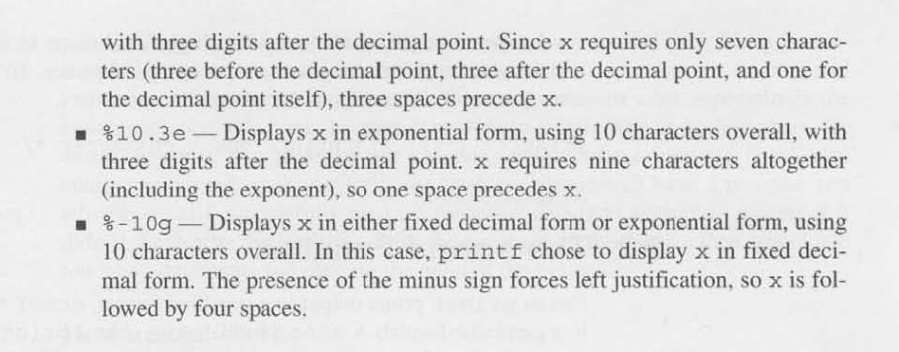
Keywords are words which hold a special meaning in the program and cannot be used as an identifier.

int a; -> this is **declaration**

int a = 10; -> this is **initialization** as we assign a value as the variable is declared

%d, %f, %g, %e is called as conversion specifiers that are to be used in printf to display values, as they convert the values from binary to the required format mentioned by the letter.

A conversion specification is of the form %m.pX where m is the **minimum field width** and p is the **precision** which depends on the **conversion specifier** i.e., X.

For example, if there is a number 123 then using %4d would right justify it like .123 (the full stop represents the space) and %-4d would left justify it like 123.

Just like Python, the C language has escape sequences which are used to represent characters which otherwise have special meaning in the language.

In C,

1. \a – audible beep
2. \t - tab
3. \b – moves cursor back one position
4. \n – new line

We don’t use \n at the end of scanf.