

011 - C IDENTIFIERS

C identifiers represent the name in the C program, for example, variables, functions, arrays, structures, unions, labels, etc.

An identifier is a collection of alphanumeric characters that begins either with an alphabetical character or an underscore, which are used to represent various programming elements such as variables, functions, arrays, structures, unions, labels, etc.

There is a total of 63 alphanumeric characters that represent the identifiers that includes 26 lowercase alphabetical characters, 26 uppercase alphabetical characters, 10 numerical digits and an underscore.

Rules for constructing C identifiers :-

The first character of an identifier should be either an alphabet or an underscore, and then it can be followed by any of the character, digit, or underscore.

It should not begin with any numerical digit.

Identifiers are case sensitive.

Commas or blank spaces cannot be specified within an identifier.

Keywords cannot be represented as an identifier.

The length of the identifiers should not be more than 31 characters.

Identifiers should be written in such a way that it is meaningful, short, and easy to read.

TYPES OF IDENTIFIERS IN C :-

INTERNAL IDENTIFIERS

If the identifier is not used in the external linkage, then it is known as an internal identifier.

The internal identifiers can be local variables.

EXTERNAL IDENTIFIERS

If the identifier is used in the external linkage, then it is known as an external identifier.

The external identifiers can be function names, global variables.
