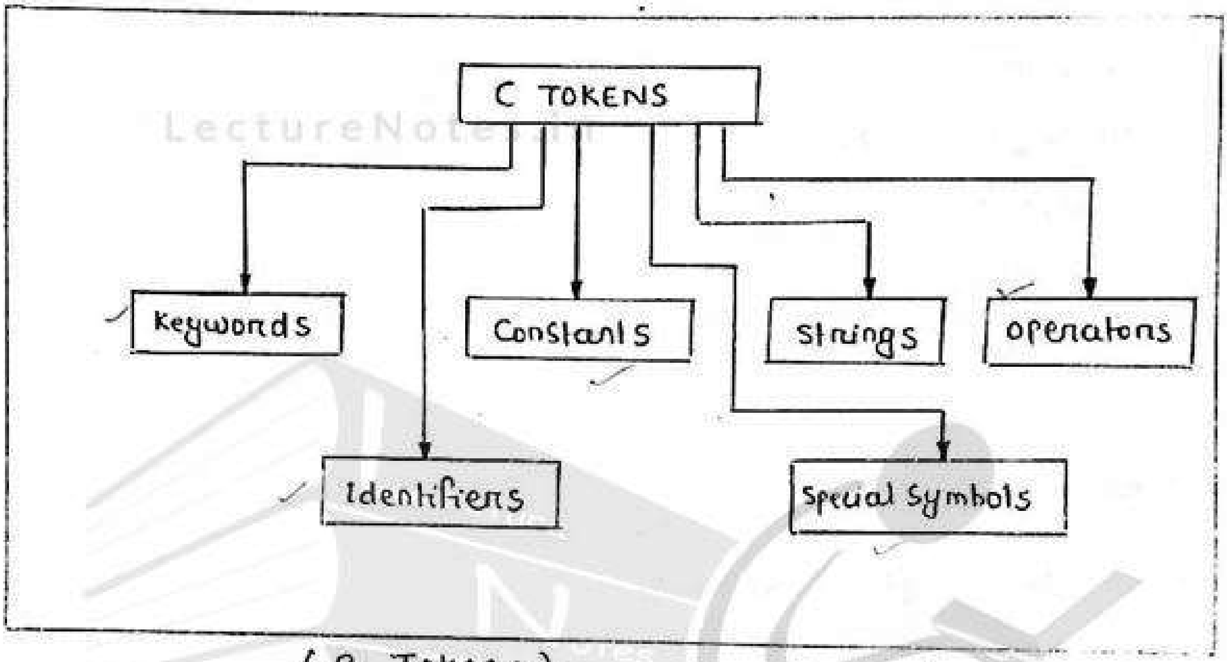
C TOKENS :

→ In a passage of text, individual words and punctuation marks are called tokens. Similarly in a C program the smallest individual units are known as C tokens.



(C Tokens)

Keywords :

→ There are certain reserved words called keyword that have Standard, pre-defined meanings in C. (32)

→ These keywords can be used only for their intended purpose. They cannot be used as programmer-defined identifiers.

auto
break
Case
char
Const
Continue
default
do

double
else
enum
extern
float
for
goto
if

int
long
register
return
Short
signed
Sizeof
Static

struct
Switch
typedef
Union
unsigned
void
volatile
while

(ANSI C keywords)

Assignment : write a program to illustrate the declaration, assignment and initialization of variable.

```
#include <stdio.h>
#include <conio.h>
void main()
 {
   clrscr();
   int a, b;  /* Declaration of variables */
   double c;
   unsigned e;
/* initialization of variables */
   int i = 100;
   long int j = 876543210L;
/* use of assigment of variables */
   a=b=400;
   c = 3.123456;
   e = 64231U;
/* Display the values */
   printf ( "a = %d\n", a);
   printf ( "b = %d\n", b);
   printf ( "c = %lf\n", c);
   printf ( "e = %u\n", e);
   printf ( "i = %d\n", i);
   printf ( "j = %d\n", j);
}
```

output :

```
a = 400
b = 400
c = 3.123456
e = 64231
i = 100
j = 876543210
```

Assignment : what is C tokens ?

→ In a passage of text, individual words and punctuation marks are called tokens.

→ Similarly in a C program, the smallest individual units are called tokens.

→ C tokens includes ; keywords, constants, identifiers, strings, special symbols and operators.

Assignment : what is keyword ?

→ There are certain reserved words in C called keywords, that have standard, predefined meanings. 32 keywords are available in C.

→ Example :

auto
break
Case
char
Constant
continue
default
do
static
typedef
void

double
else
enum
extern
float
for
goto
if
struct
union
volatile

int
long
return
register.
short
signed
sizeof
switch
unsigned
while

Assignment : what do you mean by scope of a variable ?

→ The scope of a variable determines the area of the program where that variable is valid. It depends on where it is declared. variable is of 2 types.

→ 1. Global variable : Declared outside the main function. Its scope is throughout the program. These are automatically initialized to zero.

→ 2. Local variable : Declared inside the main function. Its scope is limited to the function in which it is declared.

Lesson Number : 11

Identifiers :

→ Identifiers refers to the names of variables, functions and arrays.

→ They are user-defined names. Consisting of sequence of letters and digits. with the letter as the first character.

→ Lowercase letters are preferred. However, the uppercase letters are also permitted.

→ The under-score (_) symbol can also be used as an identifier. In general underscore is used as a link between two words in long identifiers.

Rules for identifiers :

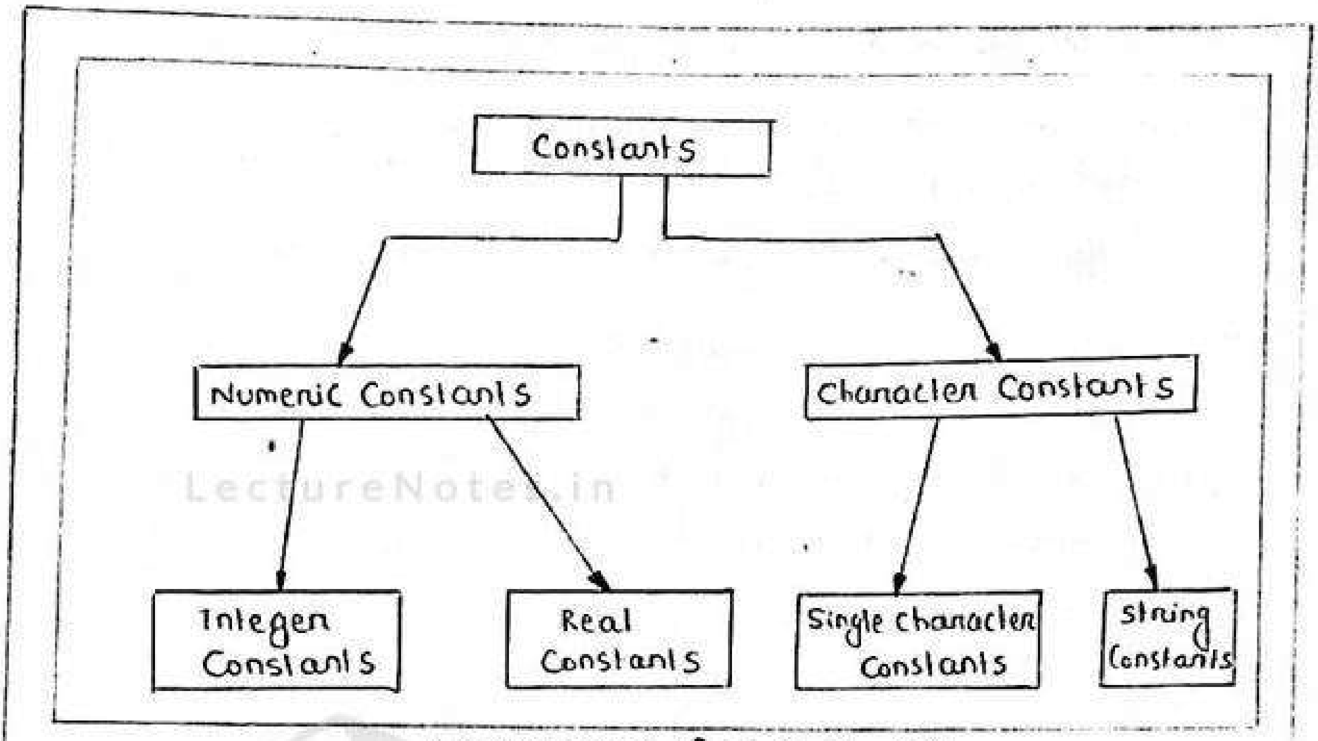
1) First character must be an alphabet or underscore.
2) Must Consists of only letters, digits or underscore.
3) Only first 31 characters are significant.
4) Cannot use a keyword
5) Must not contain white space.

valid identifier	Invalid identifier
Count	1count
Test23	hi! there
High_Score	High..Score

Constants :

→ Constants in C refers to fixed values that do not change during the execution of a program.

→ C supports several types of constants, as illustrated in the figure in the next page.



(Basic types of C Constants)

Integer Constants :

→ An integer Constant refers to a sequence of digits

→ There are three types of integers namely, decimal integer, octal integer and hexadecimal integer.

- Decimal integers consists of a set of digits 0 through 9, Preceded by an optional – or + sign.

  Examples : 123, –321, 0, 654321, +78 etc.

- An octal integer Constant consists of any combination of digits from the set 0 through 7, with a leading 0.

  Examples : 037, 0, 0453, 0551 etc.

- A sequence of digits preceded by 0X or 0x is Considered as hexadecimal integer. They may also include aphabets A through F or a through f. The letter A through F represents the numbers 10 through 15.

  Examples : 0x2, 0x9F, 0xbcd, 0x etc.

→ By default the type of an integer constant is int.

→ But if the value of integer constant (size = 2 bytes, range = -32768 to 32767) exceeds the range of values represented by int type, the type is taken to be unsigned int or long int.

→ we can also explicitly mention the type of the constant by suffixing it with l or L for long (size = 4 bytes, range = -2147483648 to 2147483647), u or U for unsigned (size = 2byte, range = 0 to 65535), ul or UL for unsigned long (size = 4 byte, range = 0 to 4294967295).

→ Example : 6453 : integer constant of type int.

4523872QUL or 4523872Qul : Integer constant of type unsigned long int.

6655U or 6655u : Integer constant of type unsigned int.

→ Representation of integer constant :

```
#include <stdio.h>
# include <conio.h>
void main()
 { printf (" Integer values\n\n");
   printf (" %d %d %d\n", 32767, 32767+1, 32767+10);
   printf ("\n");
   printf (" Long integer values\n\n");
  printf ("%d %d %d \n", 32767L, 32767L+1L, 32767L+10L);
 }
```

output :

```
Integer values
32767 -32768 -32759
Long integer values
 32767 32768 32777
```

Real Constants :

→ Real constants are often known as <u>floating point constants</u>.

→ Many parameter or quantities are defined not only in integers but also in real numbers. For example length, height, price, distance etc. are measured in real numbers.

→ <u>Example</u> : 0·0083, 3·14, −0·75, +247·0 etc.

→ A real number may also be expressed in exponential (scientific) notation.

→ For example value 215·65 may be written as 2·1565e2 in exponential notation. e2 means multiply by 10^2. The general form is ; $\boxed{\text{mantissa e exponent}}$ or $\boxed{\text{mantissa E exponent}}$

The mantissa is either a real number expressed in decimal notation or an integer. The exponent is an integer number with an optional plus or minus sign. The letter e separating the mantissa and the exponent can be written in either lowercase or uppercase.

→ <u>Example</u> : 0·65e4 12e−2 1·5e+5 3·18E3 −1·2E−1 etc.

→ 2·3e5 : floating point constant of type double.

2·4e-9l or 2·4e-9L : floating point constant of type long double.

3·52f or 3·52F : floating point constant of type float.

<u>Character Constants :</u>

1) Single character Constants:

→ In a single character constant there is a single character that is enclosed within single quotes.

→ some valid ~~sig~~ single character constants are ;

'9' 'D' '$' ' ' '#' etc.

2) String constants :

→ A string constant has zero, one or more than one character enclosed within a double quote (" ").

→ Example : "Mamata"

"567" .

"8"

" "

"A" etc.

Backslash Character Constant :

→ C supports some special backslash character constants that are used in output functions .

→ In Backslash character constant, each one of them represents one character, although they consists of two characters. These Combinations are known as escape sequences.

Constant	Meaning
'\a'	audible alert
'\b'	Back space
'\f'	form feed
'\n'	new line
'\t'	Horizontal tab
'\0'	null
'\"'	double quote
'\\'	backslash
'\v'	vertical tab

(Backslash character Constants)

Symbolic Constant :

→ If we want to use a Constant several times then we can provide it a name. For example, if we have to use the Constant 3.14159265 at many place in our program, then we can give it a name PI and use this name instead of writing the Constant value as everywhere. These types of Constants are called Symbolic Constants or named Constants.

→ A symbolic Constant is a name that substitutes for a sequence of characters. The characters may represent a numeric constant, a Character Constant or a string constant.

→ These Constants are generally defined at the beginning of the program as ;

```
#define name value
```

Here name is the symbolic name for the Constant, and is generally written in uppercase letters. 'value' can be numeric, character or string constant.

→ Some examples of symbolic Constants are ;

```
#define MAX 100
#define PI 3.14159625
#define CH 'y'
#define NAME "Mamata" etc.
```

→ In the program, these names will be replaced by the Corresponding values.

→ The symbolic Constants or named constants improve the readability and modifiability of the program.

Assignment : what is the output of the following program.

```
void main( )
 {
   printf (" Integer values\n\n");
   printf (" %d %d", 32767 , 32767 + 1);
   printf ("\n");
   printf (" %d %d %d", 32767L, 32767L + 1L , 32767 L +10L);
 }
```

output :

```
Integer values
32767 ,-32768
32767 , 32768 , 32777
```

Assignment : what is backslash character constant ?

→ C supports some backslash character constants that ared used in output functions.

→ In backslash character constant each one of them represents one character although they consists of two characters.

→ Example : '\n' → Newline

'\b' → back space

'\0' → Null

'\\' → backslash etc.

Assignment : write the syntax for defining the symbolic constant. what is the advantage of using symbolic constant.

→ Syntax : #define NAME value

Example : # define PI 3.14159625

→ The symbolic constants or named constants improve the readability and modifiability of the program.

Assignment : what is an identifier ? write the rules for naming identifier ?

→ Identifier refers to the name of variables, constants, functions and arrays.

→ They are user-defined names, consisting of sequence of letters and digits, with the letter as the first character.

→ Lowercase letters and uppercase letters both are permitted

→ The under-score (_) symbol can also be used as an identifier.

→ Example : Count
Test234
High_score etc.

Rules for Identifier :

1) First character must be an alphabet or underscore.

2) Must consists of only letters, digits or underscore.

3) only first 31 characters are significant.

4) cannot use a keyword.

5) Must not contain white space.

Assignment : what is constant ? Name the various types of constants available in C programming

→ Constants in C refers to the fixed values that do not change during the execution of the program.

→ various types of constants available in C are ;

1. Integer constants
2. Real constants
} Numeric Constants

3. Single character constants
4. String Constants
} Character constants