

### Constants and Literals

- Like a variable, a constant is a data storage location used by the users program.
- Unlike a variable, the value stored in a constant can’t be changed during program execution.
- C has two types of constants, each with its own specific uses.
  - **Literal Constants**
  - **Symbolic Constants**

#### Literal Constants

- 0 and ‘R’ are the examples for literal constant:
- int count = 20;
- char name= ‘R’;

#### Symbolic Constants

- A symbolic constant is a constant that is represented by a name (symbol) in the program.
  1. To define a symbolic constant, **#define** directive is used as follows:
     ```c
     #define CONSTNAME literal
     ```
     For example
     ```c
     #define PI 3.14159
     Area = PI * (radius) * (radius);
     ```
  2. To define a symbolic constant, Using **const** keyword.
     ```c
     const int PI 3.14159;
     Area = PI * (radius) * (radius);
     ```

```c
#include <stdio.h>
#define LENGTH 10
#define WIDTH  5
#define NEWLINE '\n'

void main() {
    int area;

    area = LENGTH * WIDTH;
    printf("value of area : %d", area);
    printf("%c", NEWLINE);
}
```

### Literals
- The constants refer to fixed values that the program may not alter during its execution.
- These fixed values are also called literals.
- Constants can be of any of the basic data types like
  - an integer constant,
  - a floating constant,
  - a character constant, or
  - a string literal.
- There are also enumeration constants as well.

### Integer literals

- An integer literal can be a decimal, octal, or hexadecimal constant.
- A prefix specifies the base or radix: 0x or 0X for hexadecimal, 0 for octal, and nothing for decimal.
- An integer literal can also have a suffix that is a combination of U and L, for unsigned and long, respectively.
- The suffix can be uppercase or lowercase and can be in any order.

#### Decimal Constant
- Allowed digits 0 to 9
- First digit must not be 0.
- eg. 91, 900, 100 are valid decimal constant
- eg. 091, 009, 0100 are invalid decimal constant

#### Octal Constant
- Allowed digits 0 to 7
- First digit must be 0.
- eg. 017, 0100, 016 are valid octal constant
- eg. 019, 018, 150 are invalid octal constant

#### Hexadecimal Constant
- Allowed digits 0 to 9, A, B, C, D, E, F
- First two characters must be 0x or 0X.
- eg. 0x60, 0x1AB, 0x10A are valid Hexadecimal constants
- eg. 0xx60, 01AB, 0x10G are invalid Hexadecimal constants

```
212          /* Legal */
215u         /* Legal */
0xFeeL       /* Legal */
078          /* Illegal: 8 is not an octal digit */
032UU        /* Illegal: cannot repeat a suffix */
```
Example

85 /* decimal */
0213 /* octal */
0x4b /* hexadecimal */
30 /* int */
30u /* unsigned int */
30l /* long */
30ul /* unsigned long */

Floating-point literals

- A floating-point literal has an integer part, a decimal point, a fractional part, and an exponent part.
- The user can represent floating point literals either in decimal form or exponential form.
- While representing using decimal form, the user must include the decimal point, the exponent, or both and while representing using exponential form; he must include the integer part, the fractional part, or both.
- The signed exponent is introduced by e or E.

3.14159 /* Legal */
314159E-5L /* Legal */
510E /* Illegal: incomplete exponent */
210f /* Illegal: no decimal or exponent */
.e55 /* Illegal: missing integer or fraction */

Character literals

- Character literals are enclosed in single quotes, e.g., ‘x’ and can be stored in a simple variable of char type.
- A character literal can be a plain character (e.g., ‘x’), an escape sequence (e.g., ‘\t’), or a universal character (e.g., ‘\u02C0’).
- There are certain characters in C when they are preceded by a backslash they will have special meaning and they are used to represent like newline (\n) or tab (\t).
- List of Escape Sequence are -

<table>
<thead>
<tr>
<th>Escape sequence</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>\ character</td>
</tr>
<tr>
<td>'</td>
<td>’ character</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot; character</td>
</tr>
<tr>
<td>?</td>
<td>? character</td>
</tr>
</tbody>
</table>
### String literals

- String literals or constants are enclosed in double quotes " ".
- A string contains characters that are similar to character literals: plain characters, escape sequences, and universal characters.
- The users can break a long line into multiple lines using string literals and separating those using whitespaces.
- String literals or constants are appended with the null character ‘\0’. It indicates the end of the string. It is used while string processing.
- ‘A’ is character constant.
- “A” is a string constant, it has two character ‘A’ and ‘\0’.