C Constants

- C Constants are also like normal variables. But, only difference is, their values can not be modified by the program once they are defined.
- Constants refer to fixed values. They are also called as literals
- Constants may be belonging to any of the data type.

- Syntax:

const data_type variable_name; (or) const data_type *variable_name;

Types of C constant:

1. Integer constants
2. Real or Floating point constants
3. Octal & Hexadecimal constants
4. Character constants
5. String constants
6. Backslash character constants

S.no	Constant type	data type	Example
1	Integer constants	int unsigned int long int long long int	53, 762, -478 etc 5000u, 1000U etc 483,647 2,147,483,680
2	Real or Floating point constants	float doule	10.456789 600.123456789
3	Octal constant	int	013 /* starts with 0 */
4	Hexadecimal constant	int	0×90 /* starts with 0x */
5	character constants	char	‘A’ , ‘B’, ‘C’
6	string constants	char	“ABCD” , “Hai”

Rules for constructing C constant:

1. Integer Constants in C:

- An integer constant must have at least one digit.
- It must not have a decimal point.
- It can either be positive or negative.
- No commas or blanks are allowed within an integer constant.
- If no sign precedes an integer constant, it is assumed to be positive.
- The allowable range for integer constants is -32768 to 32767.

2. Real constants in C:

- A real constant must have at least one digit
- It must have a decimal point
- It could be either positive or negative
- If no sign precedes an integer constant, it is assumed to be positive.

- No commas or blanks are allowed within a real constant.

3. Character and string constants in C:

- A character constant is a single alphabet, a single digit or a single special symbol enclosed within single quotes.
- The maximum length of a character constant is 1 character.
- String constants are enclosed within double quotes.

4. Backslash Character Constants in C:

- There are some characters which have special meaning in C language.
- They should be preceded by backslash symbol to make use of special function of them.
- Given below is the list of special characters and their purpose.

Backslash character	Meaning
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\”	Double quote
\’	Single quote
\\	Backslash
\v	Vertical tab
\a	Alert or bell
\?	Question mark
\N	Octal constant (N is an octal constant)
\XN	Hexadecimal constant (N – hex.dcml cnst)

How to use constants in a C program?

- We can define constants in a C program in the following ways.

1. By “const” keyword
2. By “#define” preprocessor directive

- Please note that when you try to change constant values after defining in C program, it will through error.

1. Example program using const keyword in C:

```
#include <stdio.h>
```

```
int main()
{
const int height = 100;             /*int constant*/
const float number = 3.14;   /*Real constant*/
const char letter = 'A';           /*char constant*/
const char letter_sequence[10] = "ABC";           /*string constant*/
const char backslash_char = '\?';                          /*special char cnst*/
printf("value of height :%d \n", height );
printf("value of number : %f \n", number );
printf("value of letter : %c \n", letter );
printf("value of letter_sequence : %s \n", letter_sequence);
printf("value of backslash_char : %c \n", backslash_char);
}
```

Output:

```
value of height : 100
value of number : 3.140000
value of letter : A
value of letter_sequence : ABC
value of backslash_char : ?
```

2. Example program using #define preprocessor directive in C:

```
#include <stdio.h>
#define height 100
#define number 3.14
#define letter 'A'
#define letter_sequence "ABC"
#define backslash_char '\?'
int main()
{
printf("value of height : %d \n", height );
printf("value of number : %f \n", number );
printf("value of letter : %c \n", letter );
printf("value of letter_sequence : %s \n",letter_sequence);
printf("value of backslash_char : %c \n",backslash_char);
}
```

Output:

```
value of height : 100
value of number : 3.140000
value of letter : A
value of letter_sequence : ABC
value of backslash_char : ?
```