Comments in C

Comments in C language are used to provide information about lines of code. It is widely used for documenting code. There are 2 types of comments in the C language.

1. Single Line Comments
2. Multi-Line Comments

Single Line Comments

Single line comments are represented by double slash \\. Let's see an example of a single line comment in C.

1. #include<stdio.h>
2. **int** main(){
3. //printing information
4. printf("Hello C");
5. **return** 0;
6. }

Output:

Hello C

Even you can place the comment after the statement. For example:

1. printf("Hello C");//printing information

Mult Line Comments

Multi-Line comments are represented by slash asterisk \\* ... \*\. It can occupy many lines of code, but it can't be nested. Syntax:

1. /\*
2. code
3. to be commented
4. \*/

Let's see an example of a multi-Line comment in C.

1. #include<stdio.h>
2. **int** main(){
3. /\*printing information
4. Multi-Line Comment\*/
5. printf("Hello C");
6. **return** 0;
7. }

Output:

Hello C

ASCII value in C

What is ASCII code?

The full form of ASCII is the **American Standard Code for information interchange**. It is a character encoding scheme used for electronics communication. Each character or a special character is represented by some ASCII code, and each ascii code occupies 7 bits in memory.

In [C programming language](https://www.javatpoint.com/c-programming-language-tutorial), a character variable does not contain a character value itself rather the ascii value of the character variable. The ascii value represents the character variable in numbers, and each character variable is assigned with some number range from 0 to 127. For example, the ascii value of 'A' is 65.

In the above example, we assign 'A' to the character variable whose ascii value is 65, so 65 will be stored in the character variable rather than 'A'.

**Let's understand through an example.**

**We will create a**[**program**](https://www.javatpoint.com/c-programs)**which will display the ascii value of the character variable.**

1. #include <stdio.h>
2. **int** main()
3. {
4. **char** ch;    // variable declaration
5. printf("Enter a character");
6. scanf("%c",&ch);  // user input
7. printf("\n The ascii value of the ch variable is : %d", ch);
8. **return** 0;
9. }

In the above code, the first user will give the character input, and the input will get stored in the 'ch' variable. If we print the value of the 'ch' variable by using %c format specifier, then it will display 'A' because we have given the character input as 'A', and if we use the %d format specifier then its ascii value will be displayed, i.e., 65.

**Output**



The above output shows that the user gave the input as 'A', and after giving input, the ascii value of 'A' will get printed, i.e., 65.

C Format Specifier

The Format specifier is a string used in the formatted input and output functions. The format string determines the format of the input and output. The format string always starts with a '%' character.

**The commonly used format specifiers in printf() function are:**

|  |  |
| --- | --- |
| **Format specifier** | **Description** |
| %d or %i | It is used to print the signed integer value where signed integer means that the variable can hold both positive and negative values. |
| %u | It is used to print the unsigned integer value where the unsigned integer means that the variable can hold only positive value. |
| %o | It is used to print the octal unsigned integer where octal integer value always starts with a 0 value. |
| %x | It is used to print the hexadecimal unsigned integer where the hexadecimal integer value always starts with a 0x value. In this, alphabetical characters are printed in small letters such as a, b, c, etc. |
| %X | It is used to print the hexadecimal unsigned integer, but %X prints the alphabetical characters in uppercase such as A, B, C, etc. |
| %f | It is used for printing the decimal floating-point values. By default, it prints the 6 values after '.'. |
| %e/%E | It is used for scientific notation. It is also known as Mantissa or Exponent. |
| %g | It is used to print the decimal floating-point values, and it uses the fixed precision, i.e., the value after the decimal in input would be exactly the same as the value in the output. |
| %p | It is used to print the address in a hexadecimal form. |
| %c | It is used to print the unsigned character. |
| %s | It is used to print the strings. |