Escape Sequences in C++

The **escape sequences** are special non-printing characters that are used to control the printing behavior of the output stream objects (such as ‘cout’). These characters are not displayed in the output. An escape sequence is prefixed with a backslash (\) and a coded character is used to control the printing behavior. The backslash (\) is called an escape character. So the escape sequence looks like two characters.

The escape sequence is used inside a string constant or independently. These are written in single or double-quotes. The escape sequence can be inserted in any position of the string such as:

- At the beginning of the string.
- In the middle of the string.
- At the end of the string etc.

For example, the escape sequence ‘\n’ is used to insert a new line. The cursor moves from the current position on the output device to the beginning of the next line. If escape sequence ‘\n’ is inserted at the beginning of the string then the string will be printed after printing a blank line e.g.

cout<<”\nWelcome”;

This statement results in the following output:

```
Welcome
```

List of Escape Sequences

Escape Sequence	Name	Description
`\\`	Backslash	Represents a literal backslash character.
`\'`	Single Quote	Represents a literal single quote (apostrophe).
`\"`	Double Quote	Represents a literal double quote.
`\n`	Newline	Represents a newline character.
`\r`	Carriage Return	Represents a carriage return character.
`\t`	Tab	Represents a tab character.
`\b`	Backspace	Represents a backspace character.
`\f`	Form Feed	Represents a form feed character.
`\v`	Vertical Tab	Represents a vertical tab character.
`\a`	Alert (bell)	Generates an alert (bell) sound.
`\0`	Null Character	Represents the null character (ASCII value 0).

Note - The line continuation sequence (\ followed by a new-line character) is not an escape sequence. It is used in character strings to indicate that the current line of source code continues on the next line.

- The value of an escape sequence represents the member of the character set used at run time. Escape sequences are translated during preprocessing. For example, on a system using the ASCII character codes, the value of the escape sequence \x56 is the letter V.
- Use escape sequences only in character constants or in string literals. An error message is issued if an escape sequence is not recognized.
- In string and character sequences, when you want the backslash to represent itself (rather than the beginning of an escape sequence), you must use a \ backslash escape sequence. For example: **cout << "The escape sequence \\n." << endl;**

This statement results in the following output:

```
The escape sequence \n
```

▼ Example

```
// c++ program to illustrate]
// \n escape sequence
#include <iostream>
```

```
using namespace std;

int main()
{
    // Here we are using \n, which
    // is a new line character.
        cout<<"Hello\n"<<endl;
    cout<<"GeeksforGeeks"<<endl;
    return 0;
}
```

Output

```
Hello
GeeksforGeeks
```

▼ **Example** - program that displays square, the cube of a number in table form using Escape Sequences

```
#include <iostream>
using namespace std;

int main()
{
  cout<<" Number\t Square\t Cube "<<endl;
  cout<<" 1\t\t"<<1*1<<"\t\t"<<1*1*1<<endl;
  cout<<" 2\t\t"<<2*2<<"\t\t"<<2*2*2<<endl;
  cout<<" 3\t\t"<<3*3<<"\t\t"<<3*3*3<<endl;
  cout<<" 4\t\t"<<4*4<<"\t\t"<<4*4*4<<endl;
}
```

Output

```
 Number    Square        Cube
1    1    1
2    4    8
3    9    27
4    16     64
```