Line control directive

- Last Updated : 27 Jan, 2021

Line Control Directive: Whenever a program is compiled, there are chances of occurrence of some error in the program. Whenever the compiler identifies an error in the program it provides us with the filename in which the error is found along with the list of lines and with the exact line numbers where the error is. This makes it easy for us to find and rectify the error. However, what information should the compiler provide during errors in the compilation can be controlled using the **#line** directive. It manipulates the compiler to read a line so that programmer can control the line and also the line read by the file compiler. It is basically used to reset the line number.

Characteristics:

- A C/C++ source code is actually pre-processed to create a translational unit which is then compiled.
- A translational unit has all the content of header files and itself. Hence, for the compiler, it is difficult to distinguish between the file and the line is it reading, and then it will generate error messages accordingly.
- For the preprocessor, it adds line control every time and it inserts the header file into the source code.

Syntax:

Syntax 1

# line-number "file-name"

Syntax 2

# line line-number "file-name"

line-number *— Line number that will be assigned to the next code line. The line numbers of successive lines will be increased one by one from this point on.*

"file-name" *— Optional parameter that allows to redefine the file name that will be shown.*

Explanation:

- Both of the above syntaxes are the same with a minor difference.
- **Syntax-1** will be read by the compiler while **Syntax 2** will be read by the preprocessor.
- Ultimately **syntax-2** will be converted into **Syntax 1** in the translational unit.
- These syntaxes ultimately tell the compiler that the next line will have a line number as **"line-number"** and it belongs to the file **"file-name"**.
- To verify this use two built-in macros:

- **__LINE__:** It prints the line number inside the source code.
- **__FILE__:** It prints the file name of the source code.

Below is the program to demonstrate the line control:

- C++

```
// C++ program to demonstrate line control
#include <iostream>
using namespace std;

// Driver Code
int main()
{
    cout << "START" << endl;

    cout << __LINE__ << " "
         << __FILE__ << endl;

#line 67 "Binod.cpp"

    cout << __LINE__
         << " " << __FILE__
```

```
          << endl; // (67)



     // This will apply continously (68)

     cout << __LINE__ << endl; // (69)

 }
```

Output:

```
START
11 /usr/share/IDE_PROGRAMS/C++/other/1d6323581eea236f68331bcde745f2c6/1d6323581eea236f68331bcde745f2c6.cpp
68 Binod.cpp
73
```