

Inline and Macro in C++

1. Inline :

An inline function is a normal function that is defined by the **inline** keyword.

2. An inline function is a short function that is expanded by the compiler. And its arguments are evaluated only once.

3. An inline functions are the short length functions that are automatically made the inline functions without using the **inline** keyword inside the class.

Syntax of an Inline function:

```
inline return_type function_name ( parameters )
```

```
{
```

```
// inline function code
```

```
}
```

Example of an Inline function:

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

// Inline function
inline int Maximum(int a, int b)
{
    return (a > b) ? a : b;
}

// Main function for the program
int main()
{
    cout << "Max (100, 1000):" << Maximum(100, 1000) << endl;
    cout << "Max (20, 0): " << Maximum(20, 0) << endl;

    return 0;
}
```

Output:

Max (100, 1000): 1000

Max (20, 0): 20

Macro :

It is also called **preprocessors directive**. The macros are defined by the **#define** keyword. Before the program compilation, the preprocessor examines the program whenever the preprocessor detects the macros then preprocessor replaces the macro by the macro definition.

Syntax of Macro:

```
#define MACRO_NAME Macro_definition
```

Example of Macro:

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

// macro with parameter
#define MAXIMUM(a, b) (a > b) ? a : b

// Main function for the program
int main()
{
    cout << "Max (100, 1000):";
    int k = MAXIMUM(100, 1000);
    cout << k << endl;

    cout << "Max (20, 0):";
    int k1 = MAXIMUM(20, 0);
    cout << k1;

    return 0;
}
```

Output:

Max (100, 1000):1000

Max (20, 0):20

Difference between Inline and Macro in C++ :

S.NO	Inline	Macro
1.	An inline function is defined by the inline keyword.	Whereas the macros are defined by the #define keyword.
2.	Through inline function, the class's data members can be accessed.	Whereas macro can't access the class's data members.
3.	In the case of inline function, the program can be easily debugged.	Whereas in the case of macros, the program can't be easily debugged.
4.	In the case of inline, the arguments are evaluated only once.	Whereas in the case of macro, the arguments are evaluated every time whenever macro is used in the program.
5.	In C++, inline may be defined either inside the class or outside the class.	Whereas the macro is all the time defined at the beginning of the program.
6.	In C++, inside the class, the short length functions are automatically made the inline functions.	While the macro is specifically defined.
7.	Inline is not as widely used as macros.	While the macro is widely used.
8.	Inline is not used in competitive programming.	While the macro is very much used in competitive programming.
9.	Inline function is terminated by the curly brace at the end.	While the macro is not terminated by any symbol, it is terminated by a new line.