**Inline Function** is powerful concept in C++ programming language. If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.

To make any function inline function just preceded that function with **inline** keyword.

## Why use Inline function

Whenever we call any function many time then, it take a lot of extra time in execution of series of instructions such as saving the register, pushing arguments, returning to calling function. For solve this problem in C++ introduce inline function.

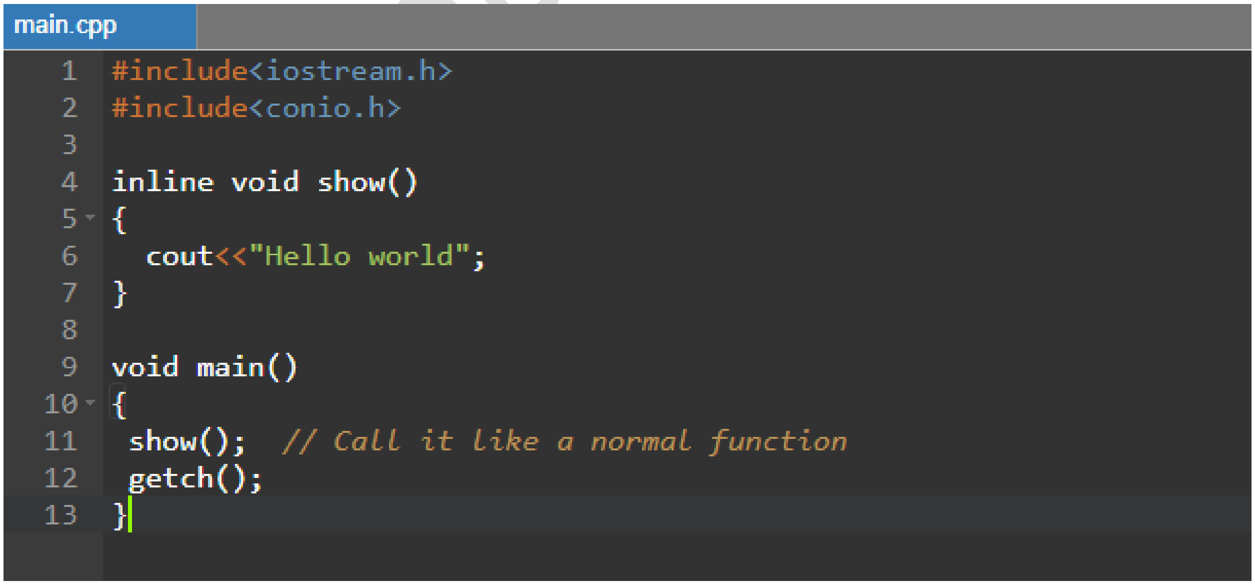
### Syntax

```
inline function_name()

{

  //function body

}
```



main.cpp

```cpp
#include<iostream.h>
#include<conio.h>

inline void show()
{
  cout<<"Hello world";
}

void main()
{
 show();  // Call it like a normal function
 getch();
}
```

## Some Important points about Inline Functions

1. We must keep inline functions small, small inline functions have better efficiency.
2. Inline functions do increase efficiency, but we should not make all the functions inline. Because if we make large functions inline, it may lead to **code bloat**, and might affect the speed too.
3. Hence, it is adviced to define large functions outside the class definition using scope resolution `::` operator, because if we define such functions inside class definition, then they become inline automatically.
4. Inline functions are kept in the Symbol Table by the compiler, and all the call for such functions is taken care at compile time.

## Limitations of Inline Functions

1. Large Inline functions cause Cache misses and affect performance negatively.
2. Compilation overhead of copying the function body everywhere in the code on compilation, which is negligible for small programs, but it makes a difference in large code bases.
3. Also, if we require address of the function in program, compiler cannot perform inlining on such functions. Because for providing address to a function, compiler will have to allocate storage to it. But inline functions doesn't get storage, they are kept in Symbol table.

## Where inline function not work?

- If inline function are recursive
- If function contain static variables.
- If return statement are exits but not return any value.