

## TIPO DE FUNCIONES

### INLINE

Indica al compilador que intente incrustar el contenido de la función en el código de llamada en lugar de ejecutar una llamada real. Para funciones pequeñas que se llaman con frecuencia que pueden marcar una gran diferencia de rendimiento.

En C++ provee el especificador de función llamado inline que cuando precede al nombre de la función hace que el compilador reemplace cualquier llamada a la función por el cuerpo actual de la función. Para utilizar esta funcionalidad, la función debe estar definida antes de ser invocada, en caso contrario, el compilador no la expandirá. Debido a esta restricción, las funciones inline son normalmente definidas en ficheros de cabecera .h.

### STATIC

Se refiere al alcance. En C significa que la función / variable solo puede usarse dentro de la misma unidad de traducción.

### CONST

Se puede anteponer el especificador const a la declaración de cualquier objeto o tipo de datos, con el fin de hacer que el objeto sea en lugar de una variable, una constante.

### FRIEND

En C ++, la función amigo permite el acceso a cualquier miembro de la clase fuera de la clase, al igual que una función miembro, la función amigo se especifica con la palabra clave amigo. Los amigos destruyen la encapsulación de los programas C ++, intentan no usar amigos y solo solicitan cuando sea apropiado

### MIEMBRO ESTATICO

Cuando la clase necesita una variable compartida por todos los objetos miembros, es mejor aplicar miembros estáticos (estáticos).

La función de miembro estático no implica este parámetro de puntero, Entonces puede usar el operador de acceso type :: scope para llamar directamente a la función miembro estática. Debido a que la función miembro estática no tiene este puntero y no necesita pasar parámetros, no es necesario usar variables miembro para llamar

### VOLATILE

Si anteponemos el calificador volatile a la declaración de un objeto, hacemos que dicho objeto pueda ser modificado por otros procesos diferentes al programa actual. Una utilidad del calificador volatile es proveer acceso a las posiciones de memoria utilizadas por procesos asíncronos, tal como manejadores de interrupciones. Los calificadores const y volatile pueden utilizarse conjuntamente o individualmente.

### NEW

El operador new asigna recursos de memoria para un objeto del tipo y tamaño especificados. En el caso de arrays, el tamaño se especifica explícitamente. En otros casos el tamaño viene definido por el propio tipo.

El operador new devuelve un puntero a un objeto del tipo especificado que referencia el espacio reservado. Si no hay espacio para crear un objeto del tipo especificado, el operador new devuelve un puntero nulo ( 0 o NULL ).

### **SOBRECARGANDO NEW**

Una clase también puede gestionar por si misma el espacio de memoria libre que se necesita para la creación de objetos. Para ello es necesario definir los operadores new y delete como funciones miembro de la clase. El operador new llamará a la función operator new, y el operador delete llamará a la función operator delete. Cuando se sobrecarguen estos operadores, la definición de estas funciones necesitan unos requerimientos que se comentan a continuación.

### **DELETE**

El operador delete destruye un objeto previamente creado por el operador new, de este modo libera la memoria que ocupaba dicho objeto. El operador delete sólo se puede aplicar a punteros si estos han sido retornados a través del operador new.

### **FUENTES**

<https://qastack.mx/programming/7762731/whats-the-difference-between-static-and-static-inline-function>

<https://programmerclick.com/article/4786834090/>

<https://analisisyprogramacionoop.blogspot.com/2014/08/c-fundamentos-basicos.html>

<https://stackoverflow.com/questions/54645193/inline-static-const-vs-static-const-variable>

<https://programmerclick.com/article/60731075603/>