¿Qué son las referencias?

Las referencias a grosso modo trabajan igual que los apuntadores en el sentido en que ambas son variables las cuales contienen dirección de la memoria, sin embargo, las referencias trabajan con una sintaxis diferente a la de los apuntadores y además las referencias no pueden ser igualadas a valores nulos como si lo son los apuntadores.

La sintaxis de las referencias nos permite poder trabajar con direcciones de memoria con la misma sintaxis que si estuviéramos trabajando con variables que no guardan direcciones de memoria, por lo cual nos permite tener una mejor legibilidad y sintaxis en el código.

Entre las principales diferencias entre punteros y referencias esta que los punteros pueden cambiar el valor de memoria que almacenan por otro, en cambio, las referencias no se pueden, ejemplo:

Int x = 2;

Int y = 3;

Int \*puntero = &x;

\*puntero = &y;

Esto no se podría hacer con referencias porque estas solo pueden tomar el valor de memoria con el cual se inicializan, también encontramos que con una referencia no podemos obtener la dirección de memoria que este almacena y finalmente no se puede hacer aritmética de puntero con las referencias.

Las referencias hacen que nuestros programas tengan una sintaxis más amigable y hay una regla muy importante a la hora de escoger entre una referencia y un apuntador y es que si vamos a necesitar inicializar la dirección de memoria a NULL, realizar aritmética de punteros o cambiar el valor al cual apunta nuestra variable, es mejor usar el tipo de dato puntero en otro caso será mejor práctica usar la referencia.

Const

Este operador nos sirve para definir tipos de datos que esperamos que sean constantes, es decir, que durante todo el código los valores iniciales que se le asignen no sean cambiados.

Primeramente, el const sirve para que a la hora de leer el código sepamos que este dato va a ser constante, además nos ayuda a que a la hora de compilar si en algún punto del código se cambió el valor del const se nos diga por consola de este error, esto sirve tanto como para variables como para parámetros, ya que sabemos que a ambos se les podría cambiar el valor con el cual se les inicializa a lo largo del código, por lo cual para impedir esto se pone const.

En el caso de retornar en una función un const esto quiere decir que luego de realizar la función y retornarle el valor de esta a una variable, esta será constante y no podrá ser modificada, por lo cual tendrá las mismas propiedades previamente habladas.