

QUALIFICATORI DI TIPO

IL TIPO CON UN QUALIFICATORE VIENE CHIAMATO **TIPO QUALIFICATO**;

CONST, VOLATILE, RESTRICT!

SEVRONO A DARE INFORMAZIONI AGGIUNTIVE AL COMPILATORE.

- **const** (C89): indica che i dati sono in sola lettura.

- **volatile** (C89): indica che un valore può cambiare tra due accessi diversi, anche se apparentemente nulla l'ha cambiato (previene ottimizzazioni del compilatore).

È una variabile che spesso modifica, quindi accedi in memoria spesso per rileggerla;

- **restrict** (C99): si utilizza solo con dichiarazioni di puntatori. Indica al compilatore che, per tutta la vita del puntatore, solo questo avrà accesso all'oggetto puntato.

ESEMPIO

Codice:

```
#include <stdio.h>
int main (void)           // main class declaration in the
{
    const volatile int local_value = 25 ; // declaring constant volatile
    int *ptr = ( int* ) &local_value ;
    printf ( " The initial value of the local_value is : %d \n ", lo
    *ptr = 195 ; // value to the pointer
    printf ( " The modified value of the local_value is: %d \n ", loc
    return 0 ;
}
```

Uscita:

```
The initial value of the local_value is : 25
The modified value of the local_value is: 195
```

Spiegazione: Nel codice sopra, puoi vedere che abbiamo dichiarato una variabile volatile costante di un tipo di dati interi con name local_value e abbiamo assegnato il valore 25 ad esso. Quindi abbiamo dichiarato il puntatore del tipo di dati interi in cui stiamo memorizzando il valore dell'indirizzo di "local_value". Inoltre, stiamo stampando il vecchio valore, quindi stiamo stampando il valore modificato sullo schermo. Questa modifica è possibile solo a causa della parola chiave volatile che abbiamo usato nella dichiarazione della variabile.