C++ volatile: A type qualifier that you can use to declare that an object can be modified in the program by the hardware.

/\* Compile code without optimization option \*/

#include <stdio.h>

int main(void)

{

    const int local = 10;

    int \*ptr = (int\*) &local;

printf("Initial value of local : %d \n", local);

    \*ptr = 100;

    printf("Modified value of local: %d \n", local);

    return 0;

}

/\*

Initial value of local : 10

Modified value of local: 100

\*/

/\* Compile code with optimization option \*/

#include <stdio.h>

int main(void)

{

    const int local = 10;

    int \*ptr = (int\*) &local;

    printf("Initial value of local : %d \n", local);

    \*ptr = 100;

    printf("Modified value of local: %d \n", local);

    return 0;

}

/\*

Initial value of local : 10

Modified value of local: 10

\*/

/\* Compile code with optimization option \*/

#include <stdio.h>

int main(void)

{

    const volatile int local = 10;

    int \*ptr = (int\*) &local;

    printf("Initial value of local : %d \n", local);

\*ptr = 100; //the keyword volatile tells the compiler –

//please do not rearrange this statement

//for any optimization purpose

    printf("Modified value of local: %d \n", local);

    return 0;

}

/\*

Initial value of local : 10

Modified value of local: 100

\*/