

In high level languages such as Java, there are functions which prevent you from accessing array out of bound by generating an exception such as `java.lang.ArrayIndexOutOfBoundsException`. But in case of C, there is no such functionality, so programmer needs to take care of this situation.

What if programmer accidentally accesses any index of array which is out of bound ?

C doesn't provide any specification which deal with problem of accessing invalid index. As per ISO C standard it is called **Undefined Behavior**.

An undefined behavior (UB) is a result of executing computer code whose behavior is not prescribed by the language specification to which the code can adhere to, for the current state of the program (e.g. memory). This generally happens when the translator of the source code makes certain assumptions, but these assumptions are not satisfied during execution.

Examples of Undefined Behavior while accessing array out of bounds

1. **Access non allocated location of memory:** The program can access some piece of memory which is owned by it.

```
// Program to demonstrate
// accessing array out of bounds
#include <stdio.h>
int main()
{
    int arr[] = {1,2,3,4,5};
    printf("arr [0] is %d\n", arr[0]);





    // arr[10] is out of bound
    printf("arr[10] is %d\n", arr[10]);
    return 0;
}
```

Output :

```
arr [0] is 1
arr[10] is -1786647872
```

It can be observed here, that `arr[10]` is accessing a memory location containing a garbage value.

2. **Segmentation fault:** The program can access some piece of memory which is not owned by it, which can cause crashing of program such as segmentation fault.



```
// Program to demonstrate
// accessing array out of bounds
#include <stdio.h>
int main()
{
    int arr[] = {1,2,3,4,5};
    printf("arr [0] is %d\n",arr[0]);
    printf("arr[10] is %d\n",arr[10]);

    // allocation memory to out of bound
    // element
    arr[10] = 11;
    printf("arr[10] is %d\n",arr[10]);
    return 0;
}
```

Output :

Runtime Error : Segmentation Fault (SIGSEGV)

Important Points:

- Stay inside the bounds of the array in C programming while using arrays to avoid any such errors.
- C++ however offers the `std::vector` class template, which does not require to perform bounds checking. A vector also has the `std::at()` member function which can perform bounds-checking.