

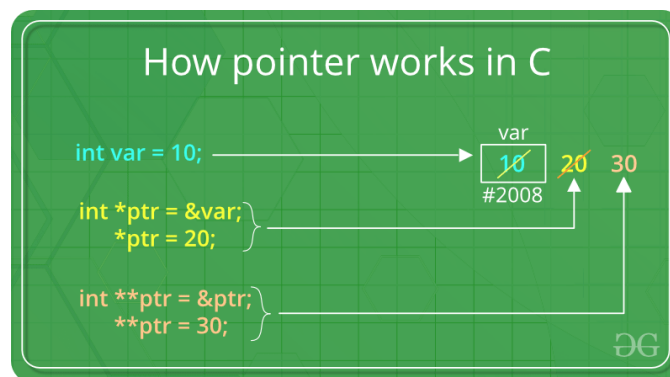
# Applications of Pointers in C++

Pointers store the address of variables or a memory location. **Syntax:**

```
datatype *var_name;
```

**Example:** pointer “ptr” holds the address of an integer variable or holds the address of memory whose value(s) can be accessed as integer values through “ptr”

```
int *ptr;
```



## Features of Pointers:

1. Pointers save memory space.
2. Execution time with pointers is faster because data are manipulated with the address, that is, direct access to memory location.
3. Memory is accessed efficiently with the pointers. The pointer assigns and releases the memory as well. Hence it can be said the Memory of pointers is dynamically allocated.
4. Pointers are used with data structures. They are useful for representing two-dimensional and multi-dimensional arrays.
5. An array, of any type, can be accessed with the help of pointers, without considering its subscript range.
6. Pointers are used for file handling.
7. Pointers are used to allocate memory dynamically.

8. In C++, a pointer declared to a base class could access the object of a derived class.  
However, a pointer to a derived class cannot access the object of a base class.

**Uses of pointers:**

1. To pass arguments by reference
2. For accessing array elements
3. To return multiple values
4. Dynamic memory allocation
5. To implement data structures
6. To do system-level programming where memory addresses are useful

**Drawbacks of Pointers:**

- If pointers are pointed to some incorrect location then it may end up reading a wrong value.
- Erroneous input always leads to an erroneous output
- Segmentation fault can occur due to uninitialized pointer.
- Pointers are slower than normal variable
- It requires one additional dereferences step
- If we forgot to deallocate a memory then it will lead to a memory leak.