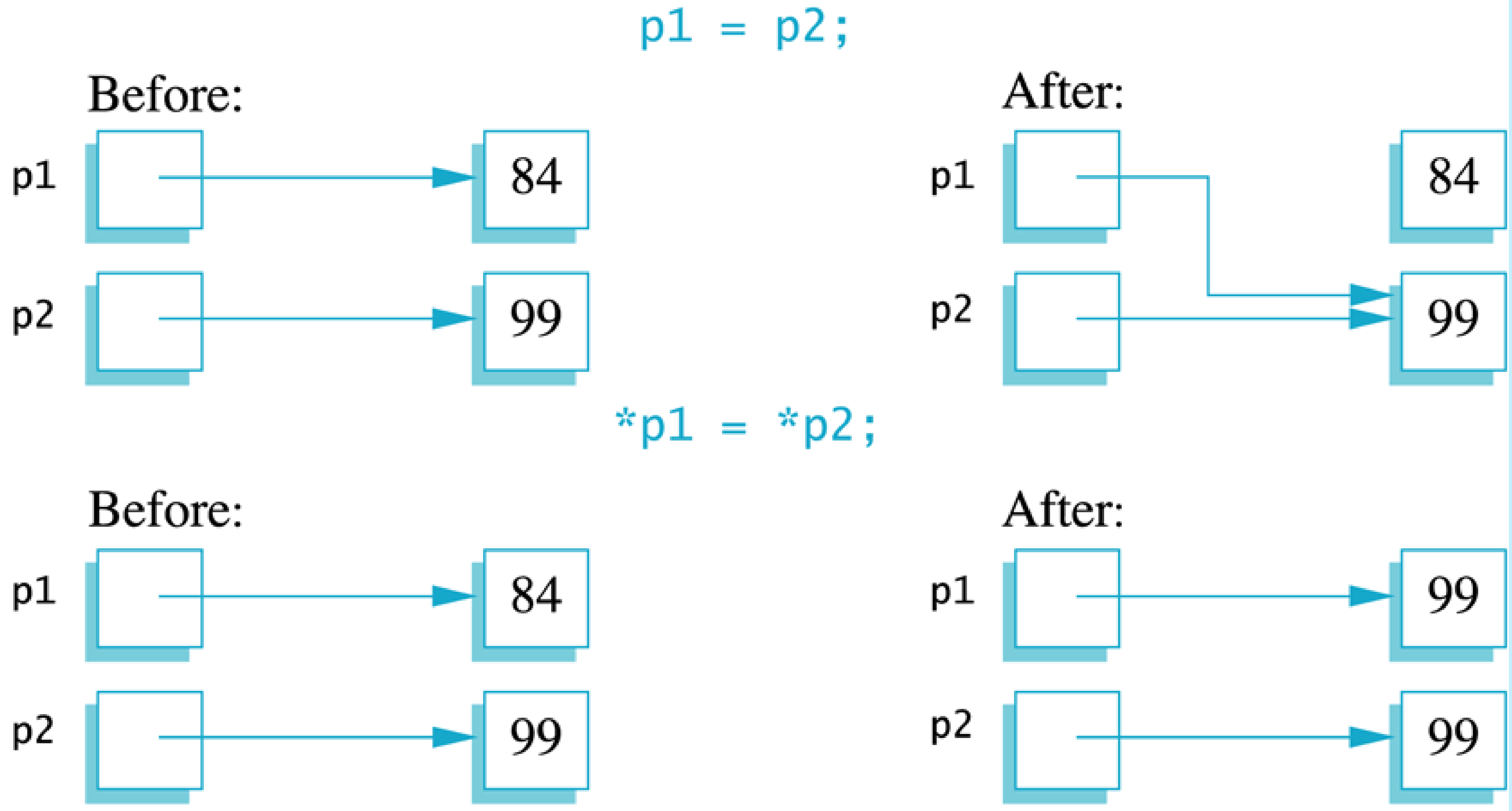
Pointer Assignment

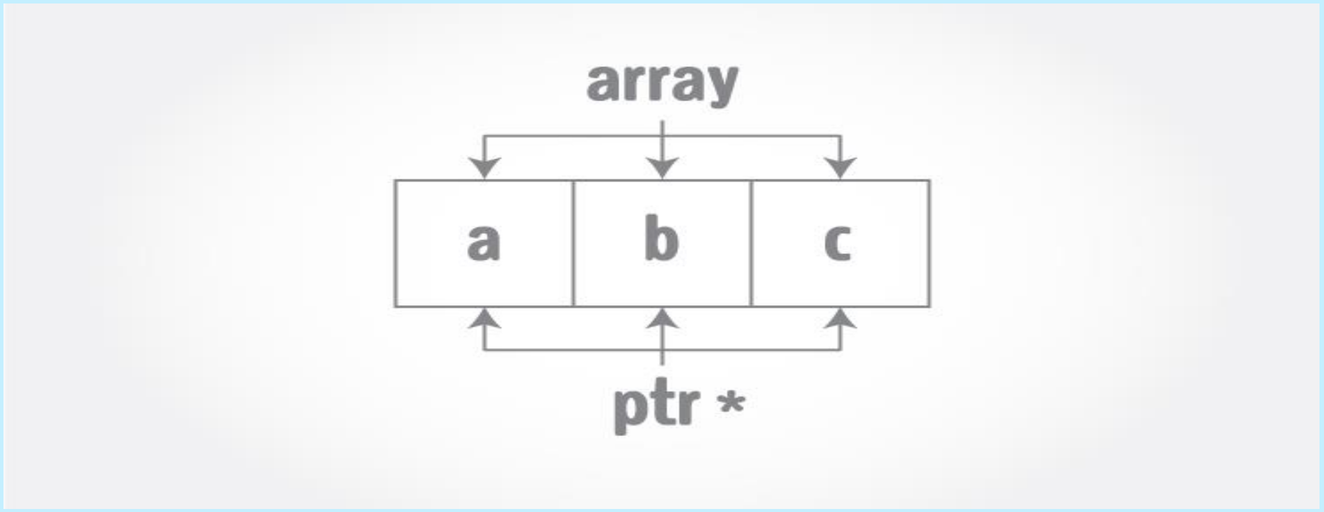
- The assignment operator = is used to assign the value of one pointer to another
  - Example: If p1 still points to v1 (previous slide) then
    p2 = p1;
  - causes *p2, *p1, and v1 all to name the same variable
- Some care is required making assignments to pointer variables
  - p1= p2; // changes the location that p1 "points" to
  - *p1 = *p2; // changes the value at the location that p1 "points" to

Uses of the Assignment Operator



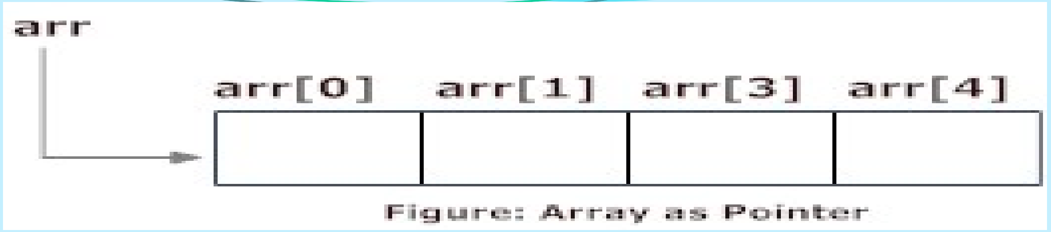
C++ Pointers and Arrays

- In this article, you'll learn about the relation between arrays and pointers, and use them efficiently in your program.



- Pointers are the variables that hold address. Not only can pointers store address of a single variable, it can also store address of cells of an array.

For example:



Int *ptr;

int a[5];

Ptr = &a[2]; // &a[2] is the address of third element of a[5].

- Suppose, pointer needs to point to the fourth element of an array, that is, hold address of fourth array element in above case.

- Since ptr points to the third element in the above example, ptr + 1 will point to the fourth element.

- You may think, ptr + 1 gives you the address of next byte to the ptr. But it's not correct.

- This is because pointer ptr is a pointer to an **int and size of int is fixed for a operating system (size of int is 4 byte of 64-bit operating system).** Hence, the address between ptr and ptr + 1 differs by 4 bytes.

- If pointer ptr was pointer to char then**, the address between ptr and ptr + 1 would have differed by 1 byte since size of a character is 1 byte.**

Example 4: C++ Pointers and Arrays
C++ Program to display address of elements of an array using both array and pointers

```
#include <iostream>
using namespace std;

int main()
{
   float arr[5];
   float *ptr;

   cout << "Displaying address using arrays: " << endl;
   for (int i = 0; i < 5; ++i)
   {
      cout << "&arr[" << i << "] = " << &arr[i] << endl;
   }

   // ptr = &arr[0]
   ptr = arr;

   cout<<"\nDisplaying address using pointers: "<< endl;
   for (int i = 0; i < 5; ++i)
   {
      cout << "ptr + " << i << " = "<< ptr + i << endl;
   }


   return 0;
}
```

Output of Eample 4:

```
Displaying address using arrays:
&arr[0] = 0x7fff5fbff880
&arr[1] = 0x7fff5fbff884
&arr[2] = 0x7fff5fbff888
&arr[3] = 0x7fff5fbff88c
&arr[4] = 0x7fff5fbff890

Displaying address using pointers:
ptr + 0 = 0x7fff5fbff880
ptr + 1 = 0x7fff5fbff884
ptr + 2 = 0x7fff5fbff888
ptr + 3 = 0x7fff5fbff88c
ptr + 4 = 0x7fff5fbff890
```

توضيح

In the above program, a different pointer ptr is used for displaying the address of array elements arr.

But, array elements can be accessed using pointer notation by using same array name arr.