

# BAAP OF ALL DSA SERIES

**NOTES**  
*on*  
**POINTERS IN C++**



**CLIC V**



alphaintern.in

## POINTERS IN C++

### What is a Pointer in C++?

A pointer is a variable that stores the memory address of another variable. Instead of holding data itself, a pointer holds the address where the data is stored in memory.

### Basic Concept of Pointers

- A **pointer** points to a **variable** in memory.
- It is declared using an asterisk \* before the pointer's name.
- The **address-of operator** & is used to get the memory address of a variable.
- The **dereference operator** \* is used to access the value stored at the address the pointer is pointing to.

### Pointer Declaration

```
type *pointer_name;
```

**type:** The type of the variable the pointer will point to (e.g., int, char, float). [simply the type of pointer variable should be the same as that variable for which we are declaring the pointer.]

**\*:** This indicates that the variable is a pointer.

**pointer\_name:** The name of the pointer

```
int *ptr; // A pointer to an integer
```

This declares a pointer ptr that will point to an int variable.

### Getting the Address of a Variable (& operator)

The **address-of operator** & is used to get the memory address of a variable.

```
int num = 10;
int *ptr = &num; // The pointer ptr holds the address of num
```

Here, ptr now stores the memory address of num.

### Dereferencing a Pointer (\* operator)

The **dereference operator** \* is used to access the value stored at the address the pointer is pointing to.

```
cout << *ptr; // Dereferencing the pointer to get the value of num, Output: 10
```

## Complete Example:

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

int main() {
    int num = 10;          // Declare an integer variable num
    int *ptr = &num;       // Declare a pointer ptr and store the address of num

    cout << "Value of num: " << num << endl;           // Output: 10
    cout << "Address of num: " << &num << endl;         // Output: Address of num (memory address)
    cout << "Pointer ptr holds the address: " << ptr << endl; // Output: Same address as &num
    cout << "Value at address held by ptr: " << *ptr << endl; // Dereference ptr, Output: 10

    return 0;
}
```

## Output:

```
Value of num: 10
Address of num: 0x61ff08
Pointer ptr holds the address: 0x61ff08
Value at address held by ptr: 10
```

## Explanation of the Code:

- `int num = 10;`: Declares an integer variable num and initializes it with 10.
- `int *ptr = &num;`: Declares a pointer ptr and stores the memory address of num in ptr.
- `&num`: This gives the address of the variable num.
- `*ptr`: This dereferences the pointer ptr and retrieves the value stored at the address ptr is pointing to, which is the value of num (10).