

# C Pointers

Topics : [C](#)

Written on [April 13, 2023](#)

A pointer in C is a variable that stores the memory address of another variable. Pointers allow you to indirectly access and manipulate data stored in memory, and they are a powerful tool for advanced programming tasks like dynamic memory allocation and passing arguments by reference.

Here's an example code snippet that shows how to declare and use pointers in C:

```
#include <stdio.h>

int main() {
    int num = 10;
    int *ptr = &num;

    printf("The value of num is: %d\n", num);
    printf("The value of ptr is: %p\n", ptr);
    printf("The value pointed to by ptr is: %d\n", *ptr);

    return 0;
}
```

Let's go over this code line by line:

- We first include the standard input/output library `stdio.h`.
- In the `main()` function, we declare an integer variable called `num` and initialize it with the value 10.
- We declare a pointer variable called `ptr` that points to the memory address of `num`. The `&` operator is used to get the address of the variable `num`.
- We use the `printf()` function to print the value of `num`, the memory address stored in `ptr`, and the value pointed to by `ptr` using the `%d` and `%p` format specifiers. The `*` operator is used to access the value pointed to by `ptr`.
- Finally, we return 0 to indicate successful execution of the program.

When you run this program, it will print the following output:

```
The value of num is: 10
The value of ptr is: 0x7fff5fbff8bc
The value pointed to by ptr is: 10
```

Note that you can use pointers to dynamically allocate memory using functions like `malloc()`,

`calloc()`, and `realloc()`. You can also use pointers to pass arguments by reference to functions, allowing you to modify the original values of variables passed to the function. However, be careful when working with pointers as invalid use of pointers can lead to program crashes and other errors.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)

ARYATECHNO