

# Chapter 6 : pointer

## 7.1 Basic Concept:

A pointer is a variable that stores the memory address of another variable. It allows you to indirectly access and manipulate the value of the variable it points to.

Example:

```
#include <stdio.h>

int main() {
    int num = 5;
    int *ptr; // Pointer declaration

    ptr = &num; // Pointer initialization

    printf("Value of num: %d\n", num);
    printf("Address of num: %p\n", &num);
    printf("Value of ptr: %p\n", ptr);
    printf("Value pointed by ptr: %d\n", *ptr); // Dereferencing the
pointer

    return 0;
}
```

## 7.2 Pointer & Arrays:

Pointers and arrays are closely related in C, as arrays can be accessed using pointers.

Example:

```
#include <stdio.h>

int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int *ptr;

    ptr = arr; // Assigning the base address of the array to the pointer

    printf("Elements of the array: ");
    for (int i = 0; i < 5; i++) {
        printf("%d ", *(ptr + i)); // Accessing array elements through
pointer arithmetic
    }

    return 0;
}
```

## 7.3 Pointer & Functions:

Pointers can be used to pass variables by reference to functions, allowing the function to directly modify the value of the variable.

Example:

```
#include <stdio.h>
```

```

void square(int *numPtr) {
    *numPtr = (*numPtr) * (*numPtr); // Squaring the value using pointer
dereferencing
}

int main() {
    int num = 5;

    printf("Before square function: %d\n", num);
    square(&num); // Passing the address of num to the function
    printf("After square function: %d\n", num);

    return 0;
}

```

#### 7.4 Pointer Arithmetic:

Pointer arithmetic is the arithmetic performed on pointers to navigate through memory locations.

Example:

```

#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;

    printf("Elements of the array: ");
    for (int i = 0; i < 5; i++) {
        printf("%d ", *(ptr + i)); // Accessing array elements through
pointer arithmetic
    }

    printf("\n");

    // Incrementing the pointer
    ptr++;
    printf("Value at incremented pointer: %d\n", *ptr);

    // Decrementing the pointer
    ptr--;
    printf("Value at decremented pointer: %d\n", *ptr);

    // Adding an offset to the pointer
    ptr += 2;
    printf("Value at pointer with offset: %d\n", *ptr);

    // Subtracting an offset from the pointer
    ptr -= 1;
    printf("Value at pointer with offset subtracted: %d\n", *ptr);

    return 0;
}

```

In this example, a pointer `ptr` is initialized with the base address of the array `arr`. Pointer arithmetic is used to access the elements of the array by incrementing and decrementing the pointer, as well as adding and subtracting offsets.