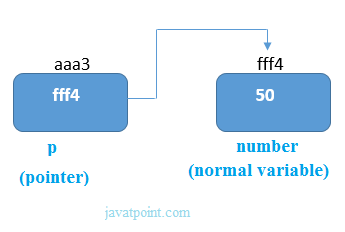
Pointer Example

An example of using pointers to print the address and value is given below.



As you can see in the above figure, pointer variable stores the address of number variable, i.e., fff4. The value of number variable is 50. But the address of pointer variable p is aaa3.

By the help of \* (**value at operator**), we can print the value of pointer variable p.

Let's see the pointer example as explained for the a

]

/>  
\\\\\\\\\\\\ “|

e.

1. #include<stdio.h>
2. **int** main()
3. {
4. **int** number=50;
5. **int** \*p;
6. p=&number;//stores the address of number variable
7. printf("Address of p variable is %x \n",p); // p contains the address of the number therefore printing p gives the address of number.
8. printf("Value of p variable is %d \n",\*p); // As we know that \* is used to dereference a pointer therefore if we print \*p, we will get the value stored at the address contained by p.
9. **return** 0;
10. }    \*p-🡪fff4--🡪50

Output

Address of number variable is fff4

Address of p variable is aaa3

Value of p variable is 50

## Advantage of pointer

1) Pointer **reduces the code** and **improves the performance**, it is used to retrieving strings, trees, etc. and used with arrays, structures, and functions.

2) It makes you able to **access any memory location** in the computer's memory.

Address Of (&) Operator

The address of operator '&' returns the address of a variable. But, we need to use %u to display the address of a variable.

1. #include<stdio.h>
2. **int** main(){
3. **int** number=50;
4. printf("value of number is %d, address of number is %u",number,&number);
5. **return** 0;
6. }

Output

value of number is 50, address of number is fff4

## NULL Pointer

A pointer that is not assigned any value but NULL is known as the NULL pointer. If you don't have any address to be specified in the pointer at the time of declaration, you can assign NULL value. It will provide a better approach.

int \*p=NULL;

Pointer Program to swap two numbers without using the 3rd variable.

1. #include<stdio.h>
2. **int** main(){
3. **int** a=10,b=20,\*p1=&a,\*p2=&b;
5. printf("Before swap: \*p1=%d \*p2=%d",\*p1,\*p2);
6. \*p1=\*p1+\*p2;  \*p1-🡪&a--🡪10…..\*p2-🡪&b--🡪20
7. \*p2=\*p1-\*p2;
8. \*p1=\*p1-\*p2;
9. printf("\nAfter swap: \*p1=%d \*p2=%d",\*p1,\*p2);
11. **return** 0;
12. }

Output

Before swap: \*p1=10 \*p2=20

After swap: \*p1=20 \*p2=10

Note:

In pointer variable from the adress we are accessing the data.