

# Pointer

## What is Pointer:

**Pointer** is a variable that represent memory location of a data item, such as variable or an array. It is a variable that hold the address of another variable. Pointer is most useful and strongest features of C++ that supports dynamic memory allocation. Pointer should not be used before declaration and initialization.

## Declaration of Pointer:

```
datatype * variablename;
```

Where datatype is type that can be declared before use like int,char etc., \* (asterisk) is known as **value at address** operator or **indirection** operator and variable name is user defined name. **Ex.:** `int *a;` // where a is called pointer variable of integer type.

## To work with pointer symbols used:

**\*: Value at address operator/Indirection operator/De reference operator**

**&: Address Operator**

## Program 1:

```
void main()
{
    int a=5;
    cout<<"\n Value of a is :"<<a;
    cout<<"\n Address of a is :"<<&a;
}
```

**Explanation (Program 1):** This program displays value and address of variable a, but address cannot be stored. To store address of a, we have to declare pointer variable separately. [ **Note:** You cannot store address in normal variable]

## Program 2:

```
void main()
{
    int a=5, *b;
    b=&a;
    cout<<"\n Value of a is :"<<a;
    cout<<"\n Address of a is :"<<b;
    cout<<"\n VALUE of a is :"<<*b;
    cout<<"\n Address of b is :"<<&b;
}
```

**Explanation (Program 2):** By this program, we will get the address and value of b, as \* will display the value of address that it has been stored previously.

### More Examples:

```
1) void main()
{
    int x,*y;
    x=5;
    y=&x;    //store address of x
    cout<<"\n Value of x is :"<<x;
    cout<<"\n Address of x is :"<<&x;
    cout<<"\n Address of x is :"<<y;
    cout<<"\n Value of x is :"<<*y;
    cout<<"\n Address of y is :"<<&y;
}
```

#### OUTPUT:

```
Value of x is :5
Address of x is :0x7ffd3c05e804
Address of x is :0x7ffd3c05e804
Value of x is :5
Address of y is :0x7ffd3c05e808
```

```
2) void main()
{
    int a1=5,a2=6,*a;
    a=&a1;
    cout<<"Value of a1 : "<<*a;
    a=&a2;
    cout<<"\nValue of a2 :"<<*a;
}
```

#### OUTPUT:

```
Value of a1 : 5
Value of a2 : 6
```

```

3) void main()
{
    int a=10,b,*x,*y,*z;
    x=&a;
    b=*x;
    y=&b;
    *z=15;
    cout<<"\n Address of a is : "<<x;
    cout<<"\n value of a : "<<b;
    cout<<"\n Address of b : "<<y;
    cout<<"\n Value of Z : "<<*z;
}

4) void main()
{
    int x,*p;
    p=&x;
    x=25;
    x=x+5; //30
    cout<<"\nValue of x : "<<x;
    x=*p+5; //35
    cout<<"\nValue of x : "<<x;
    *p=*p+5; //40
    cout<<"\nValue of x : "<<*p;
    cout<<"\nAddress of x"<<&*p; //address of x
}

```

**OUTPUT:**

```

Value of x : 30
Value of x : 35
Value of x : 40
Address of x : 0x7ffdc5e4f614

```

## Uses of pointers:

- 1) It directly access memory location of a variable and hence can be manipulated the way it required.
- 2) Pointer can access array elements.
- 3) It can be used as parameters to a function when the function needs to be modify the original arguments.
- 4) With the help of pointer, array and string can be passed to a function.
- 5) Data structure can be created by the pointer like linked list and binary trees.
- 6) It reduce length and complexity of a program that increases execution speed
- 7) Pointer supports allocation and de-allocation of memory.

## Pointer and Arrays:

Pointers and arrays are very closely related to each other. The name of array specifies the starting or base address of the array. In fact it is the first element of the array.

```
int a[5]={10,15,20,25,30};
```

### Accessing array elements:

#### 1) Array indexing method:

```
i)   cout<<a[0];    // show the value
ii)  cout<<&a[0];    // show the address
iii) cout<<*a;       // show the first value
iv)  cout<<a;        // show the first value's address
```

#### 2) Pointer method:

for elements: `*(a+i)` and address `(a+i)` can be used in pointer method.

**Example:**

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int a[5],m,*p,i;
    clrscr();
    for(i=0;i<5;i++)
    {
        cout<<"Enter array element:";
        cin>>a[i];
    }
    p=&a[0];
    m=*p;
    for(i=0;i<5;i++)
    {
        if(m<*p)
        {
            m=*p;
        }
    }
    p++;
}
cout<<"Biggest number in array is: "<<m;
getch();
}
```

**OUTPUT:**

```
Enter array element:5
Enter array element:25
Enter array element:99
Enter array element:1
Enter array element:215
Biggest number in array is: 215
```

//BOARD PRACTICAL TRVERSING AN ARRAY PROGRAM

**Question:** Write a program in C++ that first initializes an array of five given numbers (short /float/ double). The program must add these numbers by traversing this array with a pointer. The output should print the starting address of the array with the size of the number (in bytes) to which it points. The program must also print the sum and pointer address with addition of every number as well as the ending address.

**Program:**

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    float a[] = {5.5, 4.4, 3.3, 2.2, 1.1}, sum = 0;
    int i;

    cout<<"\n The base address of the array is : "<<a;
    cout<<"\n The base element of the array is : "<<*a;
    cout<<"\n The size of the element to which the pointer
is pointing : "<<sizeof(*a)<<" bytes";

    for(i=0; i<5 ; i++)
    {
        sum = sum + *(a+i);

        cout<<"\n The "<<(i+1)<<" element in the array is
: "<<*(a+i)<<" and its address is : "<<(a+i);

        cout<<"\n The sum of first "<<(i+1)<<" element(s)
is : "<<sum;
    }
    cout<<"\n The ending address of the array is :
"<<(a+4);
    cout<<"\n The ending element of the array is :
"<<*(a+4);
    getch();
}
```

**Sample Output:**

The base address of the array is : 0x8f23ffde  
The base element of the array is : 5.5  
The size of the element to which the pointer is pointing :  
4 bytes  
The 1 element in the array is : 5.5 and its address is :  
0x8f23ffde  
The sum of first 1 element(s) is : 5.5  
The 2 element in the array is : 4.4 and its address is :  
0x8f23ffe2  
The sum of first 2 element(s) is : 9.9  
The 3 element in the array is : 3.3 and its address is :  
0x8f23ffe6  
The sum of first 3 element(s) is : 13.2  
The 4 element in the array is : 2.2 and its address is :  
0x8f23ffea  
The sum of first 4 element(s) is : 15.4  
The 5 element in the array is : 1.1 and its address is :  
0x8f23ffee  
The sum of first 5 element(s) is : 16.5  
The ending address of the array is : 0x8f23ffee  
The ending element of the array is : 1.1

### Dynamic allocation operator:

1) new 2) delete

**new:** It is keyword used as operator to allocates the memory dynamically by the programmer. It returns a pointer storing the memory address of the allocated memory.

### Example:

```
1. int *a=new int;           //declaration and assignment
2. int *a=new int(5);        //initialize
```

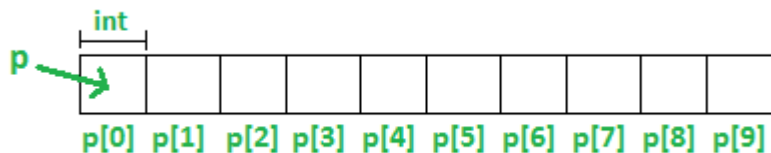
**Allocate block of memory:** new operator is also used to allocate a block (an array) of memory of type *data-type*.

```
pointer-variable = new data-type[size];
```

where size (a variable) specifies the number of elements in an array.

Example: `int *p = new int[10]`

Dynamically allocates memory for 10 integers continuously of type int and returns pointer to the first element of the sequence, which is assigned to p (a pointer). p[0] refers to first element, p[1] refers to second element and so on.



**delete:** It is used to de-allocate the memory space that has been allocated by new operator.

**Example:** `delete *a;`

**Question 1:** Write program in C++ to enter the n elements in an array and display the same using pointer.

**Question 2:** Write program a C++ program to display the elements and sum of all the elements present in that array using pointer.

**Question 3:** Write a program to initialize an array with 5 elements and display all the elements of array with their addresses.