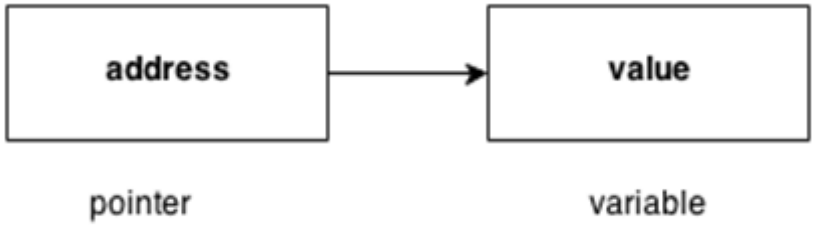
Pointers -

- The pointer in C++ language is a variable, it is also known as locator or indicator that points to an address of a value.



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) We can return multiple values from function using pointer.

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

Usage of pointer -

There are many usage of pointers in C++ language.

1) **Dynamic memory allocation -** In c language, we can dynamically allocate memory using malloc() and calloc() functions where pointer is used.

2) Arrays, Functions and Structures - Pointers in c language are widely used in arrays, functions and structures. It reduces the code and improves the performance.

Symbols used in pointer -

Symbol	Name	Description
& (ampersand sign)	Address operator	Determine the address of a variable.
* (asterisk sign)	Indirection operator	Access the value of an address.

Declaring a Pointer

- **Pointer to int – int *a;**
- **Pointer to char – char *c;**

```
#include<iostream.h>
#include<conio.h>

void main()
{
   int number  = 30;
   int *p;
   p = &number;

   cout<<"Value of number: "<<number;
   cout<<"Value of p: "<<p;
   cout<<"Value of *p: "<<*p;
}
```

/* WAP to swap two numbers without using 3rd variable*/

```
#include<iostream.h>
#include<conio.h>

void main()
{
        int a = 20,b = 10;
        *p1 = &a;
        *p2 = &b;

        cout<<"Before swap: *p1= "<<*p1;
        cout<<"Before swap: *p2= "<<*p2;

        *p1 = *p1 + *p2;
        *p2 = *p1 - *p2;
        *p1 = *p1 - *p2;

        cout<<"After swap: *p1= "<<*p1;
         cout<<"After swap: *p2= "<<*p2;
}
```

Pointer to Variable -

```
#include<iostream.h>
#include<conio.h>

void main()
{
        int n,*p;
        cout<<“Enter the number: ”;
        cin>>n;

        p = &n;

        cout<<“\nValue of n: ”<<n;
        cout<<“\nAddress of n: ”<<&n;
        cout<<“\nValue of p: ”<<p;
        cout<<“\nAddress of p: ”<<&p;
        cout<<“\nContent of p: ”<<*p;
}
```

Pointer to Array -

```
#include<iostream.h>
#include<conio.h>

void main()
{
        int arr[5],i,*p;

        cout<<"Enter the array elements: ";
        for(i=0;i<5;i++)
        {
                cin>>arr[i];
        }

        p=&arr[0];

        cout<<"Value is: "<<*p;                    //it prints the first element of array
}
```

Traversing an array using Pointer -

```
#include<iostream.h>
#include<conio.h>

void main()
{
        int arr[5],i,*p;

        cout<<"Enter the array elements: ";
        for(i=0;i<5;i++)
        {
                cin>>arr[i];
        }

        p=&arr[0];

        cout<<"\nArray Elements are:\n";
        for(i=0;i<5;i++)
        {
                cout<<" "<<*(p+i);
        }
}
```

Pointer to Function – Type - I

```
#include<iostream.h>
#include<conio.h>

void sum(int x, int y)
{
        int c = (x+y);
        cout<<"Sum is: "<<c;
}

void main()
{
        void (*fp)(int p, int q);
        fp = sum;

        fp(10,20);
}
```

Pointer to Function – Type - II

```
#include<iostream.h>
#include<conio.h>

int sum(int x, int y)
{
        int c = (x+y);
        return c;
}

void main()
{
        int (*fp)(int a, int b);
        fp = sum;
        int s = fp(10,20);
        cout<<"Value of s: "<<s;
}
```

Pointer to Structure -

```
#include<iostream.h>
#include<conio.h>

struct student
{
        int id;
        char name[10];
        float perc;
};

void main()
{
        struct student record = {1,"GGI",90.5};
        struct student *ptr;

        ptr = &record;
        cout<<"\nRecords of Students\n";
        cout<<"Id is: "<<ptr->id;
        cout<<"\nName is: "<<ptr->name;
        cout<<"\nPercentage is: "<<ptr->perc;
}
```

Constant Pointer

```
#include<iostream.h>
#include<conio.h>

void main()
{
        int arr[2] = {10,20};
        int *const ptr = &arr[0];

        cout<<"Value is: "<<*ptr;
}
```