

Pointers in C++

A pointer is a memory variable that points to the specific address in the memory pointed by another variable. We have two operators (& and *) in c++ , one for referencing the memory and other for de-referencing purpose.

Dereference operator (*):

The (*) de-reference operator. This is used to hold the address of another variable of same data type.

The syntax for declaration of pointer variable is :

```
<data-type> *ptr;
```

Or

```
<data-type>* ptr;
```

For example `int *ptr;`

Here ptr is pointer variable that is used to hold the address of another variable

Reference operator (&)

The reference (&)operator is used to return the memory address of the variable. It is a unary operator and is written as :

```
&ptr;
```

This will return the address of ptr. The & sign is used to assign the address of some other variable/object of same data type.

Pointer expression

It should be emphasized here that if we have declared a pointer variable, then it should be assigned the address of the variable which is also of type integer.

For example:

```
int x, *ptr;           then we can use the statement as:
```

```
ptr = &x;
```

ptr is assigned the address of the variable x and the data type of ptr and x is the same.

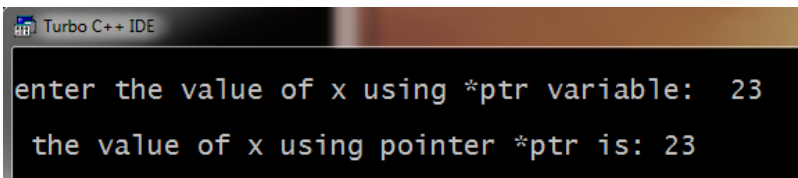
Whereas the reference variable is used to assign the address of the variable to pointer variable and that the type is same, the de-reference operator (*) is used to access the value held at the address by the variable. The following example illustrates this:

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    int x;
    int *ptr;
    ptr=&x;    //ptr is assigned the address of x
    cout<<"\nenter the value of x using *ptr variable: ";
    cin>>*ptr;
    cout <<"\n the value of x using pointer *ptr is: "<<x;
    getch();
}
```

```

#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int x;
int *ptr;
ptr=&x;    //ptr is assigned the address of x
cout<<"\nenter the value of x using *ptr variable: ";
cin>>*ptr;
cout<<"\n the value of x using pointer *ptr is: "<<x;
x++;
cout<<"\nnew val of x after increment x = "<<x;
cout<<"\nqaddress of x held in ptr = "<<ptrinclude<iostream.h>

```



The screenshot shows the Turbo C++ IDE window with the following output in the console:

```

enter the value of x using *ptr variable: 23
the value of x using pointer *ptr is: 23

```

Operations on Pointers

The value held by the pointer variables is the address of the variables or say data items.

```
*ptr=*ptr+2;
```

Abstract Classes,

Virtual Functions, Polymorphism

```
class Base          //Abstract base class
```

```
{
    public:
    virtual void show() = 0;           //Pure Virtual Function
};
class Derived:public Base
{
    public:
    void show()
    { cout << "Implementation of Virtual Function in Derived class"; }
};

int main()
{
    Base obj;           //Compile Time Error
    Base *b;
    Derived d;
    b = &d;
    b->show();
}
```