C++ POINTER OPERATORS

http://www.tutorialspoint.com/cplusplus/cpp_pointer_operators.htm

C++ provides two pointer operators, which are *a* Address of Operator & and *b* Indirection Operator *.

A pointer is a variable that contains the address of another variable or you can say that a variable that contains the address of another variable is said to "point to" the other variable. A variable can be any data type including an object, structure or again pointer itself.

The . *dot* operator and the -> *arrow* operator are used to reference individual members of classes, structures, and unions.

The Address of Operator &:

The & is a unary operator that returns the memory address of its operand. For example, if var is an integer variable, then &var is its address. This operator has the same precedence and right-to-left associativity as the other unary operators.

You should read the & operator as **"the address of"** which means **&var** will be read as "the address of var".

The Indirection Operator *:

The second operator is indirection Operator *, and it is the complement of &. It is a unary operator that returns the value of the variable located at the address specified by its operand.

The following program executes the two operations

```
#include <iostream>

using namespace std;

int main ()
{
   int  var;
   int  *ptr;
   int  val;

   var = 3000;

   // take the address of var
   ptr = &var;

   // take the value available at ptr
   val = *ptr;
   cout << "Value of var :" << var << endl;
   cout << "Value of ptr :" << ptr << endl;
   cout << "Value of val :" << val << endl;

   return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Value of var :3000
Value of ptr :0xbff64494
Value of val :3000
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js