

POINTERS

Accessing variables through pointers (Dereferencing Operators)

```
#include <stdio.h>

int main()
{
    int x=100,*ptr1; //declaration of pointer
    char y='A',*ptr2; //declaration of pointer
    ptr1=&x; //intizan of pointer
    ptr2=&y; //intizan of pointer

    //Size of Variables
    printf("Size of x is %d\n",sizeof(x));
    printf("Size of y is %d\n",sizeof(y));
    printf("Size of ptr1 is %d\n",sizeof(ptr1));
    printf("Size of ptr2 is %d\n",sizeof(ptr2));

    //address of Variables
    printf("Address of x is %X\n",&x);
    printf("Address of y is %X\n",&y);
    printf("Address of ptr1 is %X\n",&ptr1);
    printf("Address of ptr2 is %X\n",&ptr2);

    //values of Variables
    printf("Values of x =%d\n",x);
    printf("Values of y =%c\n",y);
    printf("Values of ptr1 =%ld\n",ptr1);
    printf("Values of ptr2 =%ld\n",ptr2);

    // Printing the values of variables using pointers
    printf("Value of x is %d\n",*ptr1);
    printf("Value of y is %c\n",*ptr2);
}
```

The void pointer

- ✓ Suppose we have to declare integer pointer, character pointer and float pointer then we need to declare 3 pointer variables.
- ✓ Instead of declaring different types of pointer variable it is feasible to declare single pointer variable which can act as integer pointer, character pointer.

Basics-

- ✓ In C General Purpose Pointer is called as void Pointer.
- ✓ It does not have any data type associated with it
- ✓ It can store address of any type of variable
- ✓ A void pointer is a C convention for a raw address.
- ✓ The compiler has no idea what type of object a void Pointer really points to ?

Declaration of Void Pointer:

void * pointer_name;

Example:

```
void *ptr; // ptr is declared as Void pointer
char cnum;
int inum;
float fnum;
ptr = &cnum; // ptr has address of character data
ptr = &inum; // ptr has address of integer data
ptr = &fnum; // ptr has address of float data
```

Explanation:

void * ptr;

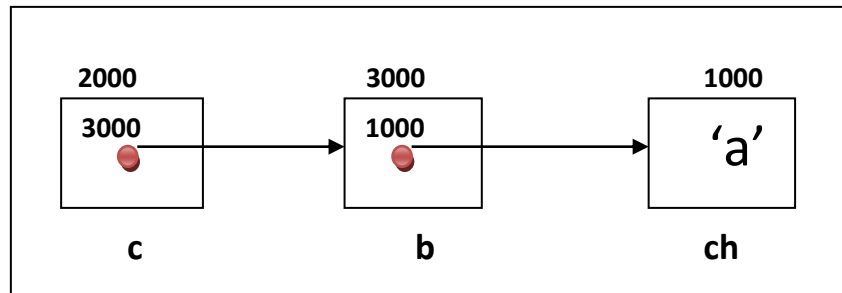
- ✓ Void pointer declaration is shown above.
- ✓ We have declared 3 variables of integer, character and float type.
- ✓ When we assign address of integer to the void pointer, pointer will become Integer Pointer.
- ✓ When we assign address of Character Data type to void pointer it will become Character Pointer.
- ✓ Similarly we can assign address of any data type to the void pointer.
- ✓ It is capable of storing address of any data type

Pointers to pointers

C allows the use of pointers that point to pointers, that these, in its turn, point to data (or even to other pointers). In order to do that, we only need to add an asterisk (*) for each level of reference in their declarations:

```
char ch;  
char *b;  
char * *c;  
ch = 'a';  
b = &ch;  
c = &b;
```

This, supposing the randomly chosen memory locations for each variable c, b and ch could be represented as:



The value of each variable is written inside each cell; Under the cells are their respective addresses in memory.

The new thing in this example is variable c, which can be used in three different levels of indirection, each one of them would correspond to a different value:

- ➔ c has type char** and a value of 3000
- ➔ *c has type char* and a value of 1000
- ➔ **c has type char and a value of 'a'