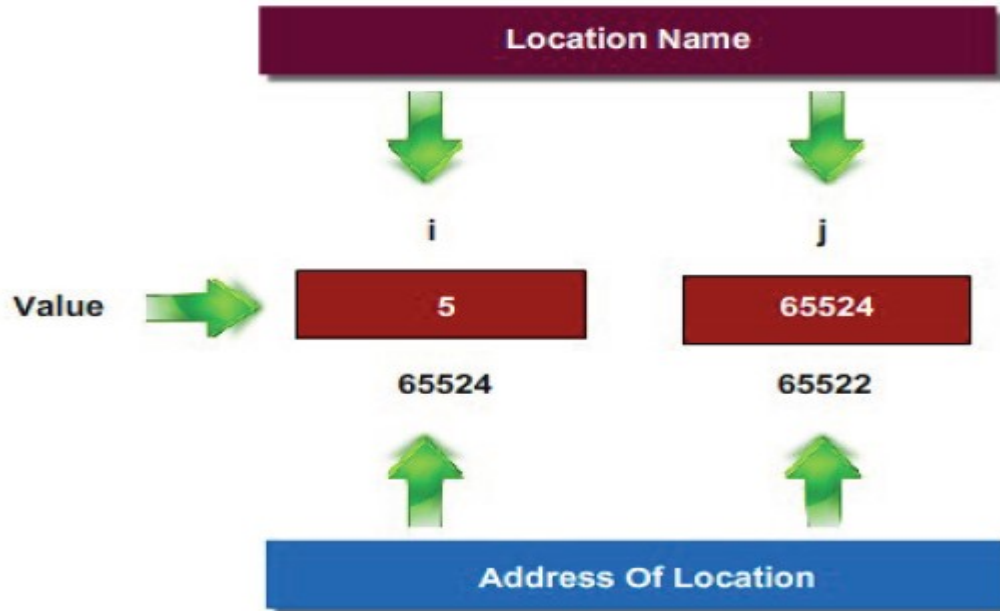


Chapter - 9 : Pointers

Pointers

A pointer is a special variable which holds the address of the variable it has pointed to.

Basic Concept of Pointer



- “i” is the name given for Particular memory location.
- Consider it's Corresponding address be 65524 and the Value stored in variable ‘i’ is 5
- The address of the variable ‘i’ is stored in another integer variable whose name is ‘j’ and which is having corresponding address 65522
- Thus, one can say that, $j = \&i$; that is $j = \text{Address of } i$.
- Here j is not ordinary variable; It is special variable and called pointer variable as it stores the address of the ordinary variable

Variable Name	Variable Value	Variable Address
i	5	65524
j	65524	65522

Address operator in C programming

- It is Denoted by ‘&’
- When used as a prefix to a variable name ‘&’ operator gives the address of that variable.
- Example : **&n** Gives address on **n**.

Example

```
#include <stdio.h>
void main()
{
    int n=10;
    printf("\nValue of n is : %d",n);
    printf("\nValue of &n is %u", &n);
}
```

Output

```
Value of n is : 10
Value of &n is : 65522
```

Using pointer Variable

Pointer variable can be defined using the (*).

```
<data type> * pointer_variable_name;
```

```
int * ptr;
```

```
float *ptr;
```

```
char *ptr;
```

Example

```
#include <stdio.h>
void main()
{
    int i=5;
    int *ptr;
    ptr=&i;
    printf("\nAddress of i is : %u", &i);
    printf("\nValue of ptr is %u", ptr);
}
```

Output

```
Address of i is : 65522
Value of ptr is : 65522
```

Invalid Use of Address Operator

- Programmer cannot use Address operator for Accessing Address of Literals.
- Only Variables have Address associated with them. **&75**
- **(a+b)** will evaluate addition of Values present in variables.
- Output of (a+b) is nothing but Literal, so one cannot use Address operator. **&(a+b)**
- Again 'a' is Character Literal, so he cannot use Address operator. **&('a')**

Pointer Assignments

- To assign an address to a pointer we'll need a new operator, the "address of" operator.
- Once assigned, the pointer will contain the "address of" the assigned variable not it's value.
- Code Example:
 - `int *xp; // declares xp as a pointer to an integer`
 - `xp = &x; // xp receives the address of 'x'`
- Pointer variables can be "assigned":
 - `int *p1, *p2;`
 - `p2 = p1;`
 - Assigns one pointer to another.
 - "Make p2 point to where p1 points".
- Do not confuse with:
 - `*p1 = *p2;`
 - Assigns "value pointed to" by p1, to "value pointed to" by p2.

Uses of the Assignment Operator With Pointer Variables

`p1 = p2;`



`*p1 = *p2;`



Example # Program to assign the address of variable to pointer, and print the data using pointer variable

```
#include <stdio.h>
void main()
{
    int n;
    int *ptr;
    n=10;
    ptr=&n;

    printf("\nValue of n is : %d", *ptr );
    printf("\nValue of &n is %u", ptr);
}
```

Output

```
Value of n is : 10
Value of &n is : 65522
```

Example # Program to assign the address of two variable to two pointer, and print the sum of data using pointer variable

```
#include <stdio.h>
void main()
{
    int a,b;
    int *ptr1, *ptr2;
    a=10;
    b=20;

    ptr1=&a;
    ptr2=&b;

    printf("\nValue of sum is : %d", *ptr1 + *ptr2 );
}
```

Output

Value of sum is : 20

Example # Program to assign the address of two variable to two pointer, and print the greatest of data using pointer variable

```
#include <stdio.h>
void main()
{
    int a,b;
    int *ptr1, *ptr2;
    a=10;
    b=20;

    ptr1=&a;
    ptr2=&b;

    if( *ptr1 > *ptr2 )
        printf("\nGreatest is : %d", *ptr1 );
    else
        printf("\nGreatest is : %d", *ptr2 );

}
```

Output

Greatest is : 20