Introduction to Pointers

MODIFIED BY

MD. MOTALEB HOSSEN MANIK

LECTURER

CSE, KUET

Concept of Variable, Value & Address

int i = 3;

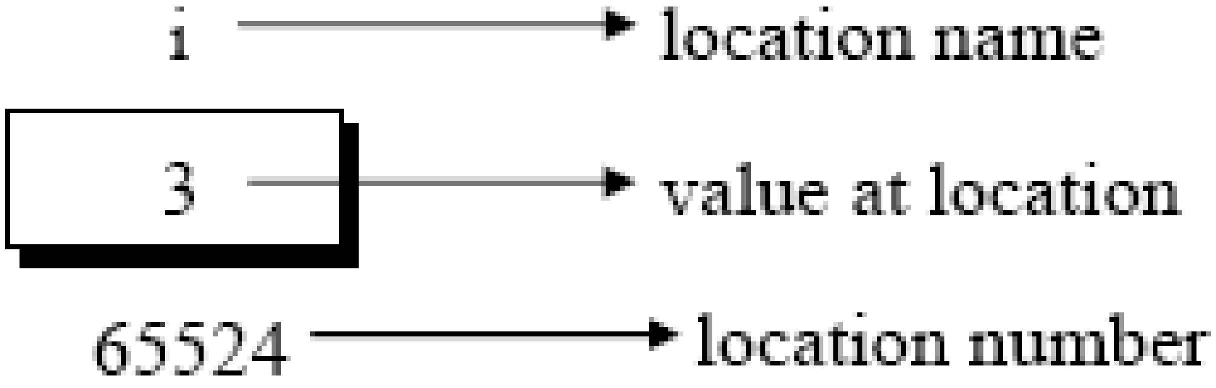
This declaration tells the C compiler to:

(a) Reserve space in memory to hold the integer value.

(b) Associate the name **i** with this memory location.

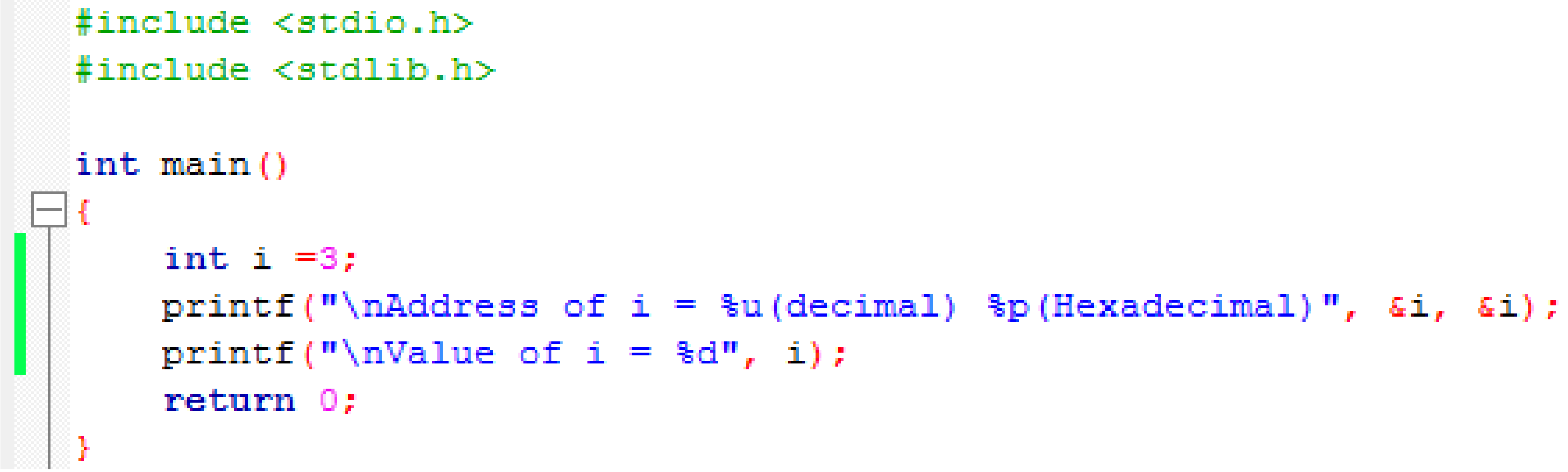
(c) Store the value 3 at this location.

Concept of Variable, Value & Address

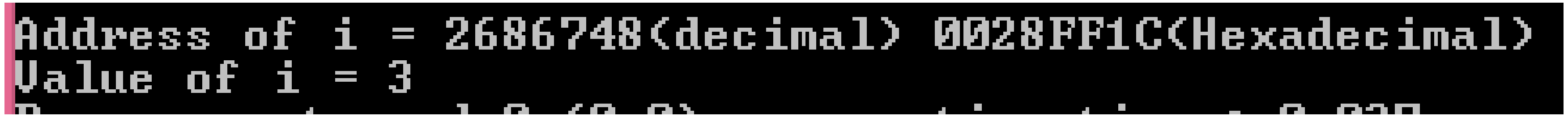


The important point is, **i**’s address in memory is a number.

Concept of Variable, Value & Address



```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i =3;
    printf("\nAddress of i = %u(decimal) %p(Hexadecimal)", &i, &i);
    printf("\nValue of i = %d", i);
    return 0;
}
```



```
Address of i = 2686748(decimal) 0028FF1C(Hexadecimal)
Value of i = 3
```

Concept of Variable, Value & Address

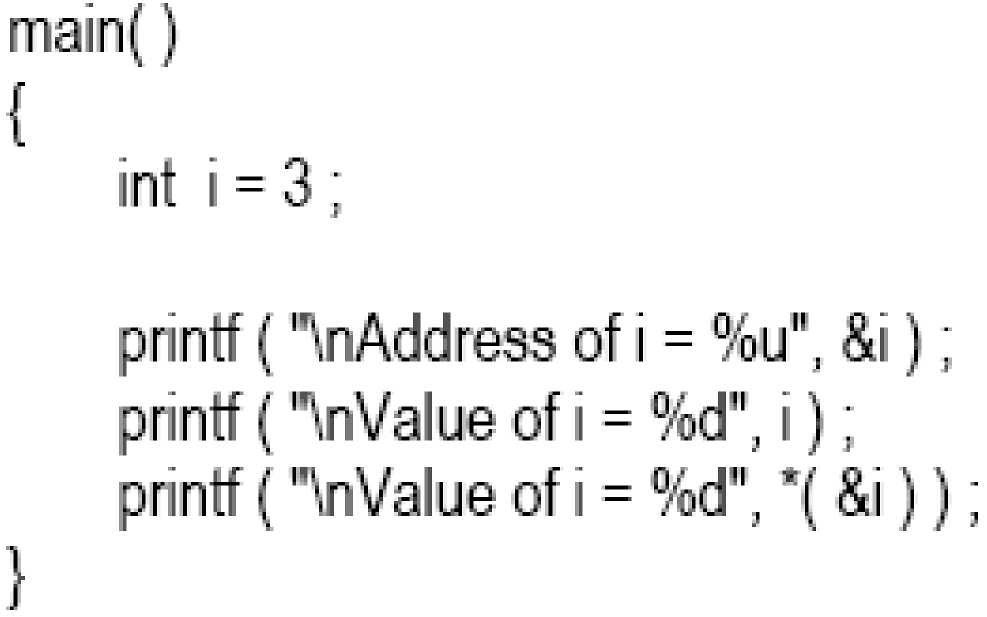
The other pointer operator available in C is ‘*****’

called ‘value at address’ operator

It gives the value stored at a particular address

The ‘value at address’ operator is also called ‘indirection’ operator

Concept of Variable, Value & Address



```
main( )
{
        int  i = 3 ;

        printf ( "\nAddress of i = %u", &i ) ;
        printf ( "\nValue of i = %d", i ) ;
        printf ( "\nValue of i = %d", *( &i ) ) ;
}
```

Concept of Pointer

The expression **&i** gives the address of the variable **i**

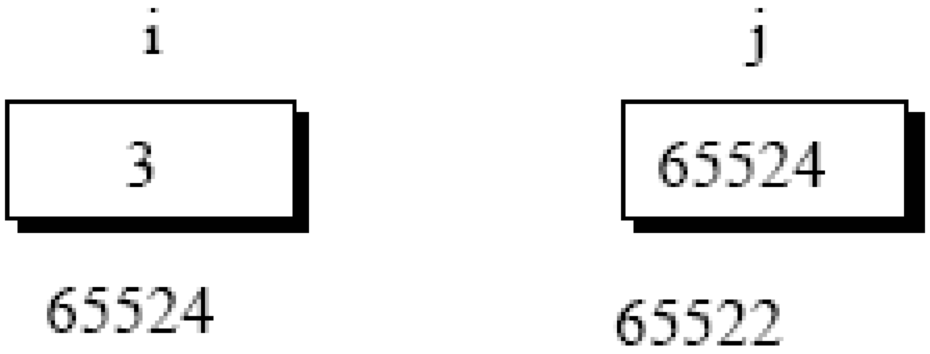
This address can be collected in a variable

j = &i ;

But remember that **j** is not an ordinary variable like any other integer variable

It is a variable that contains the address of other variable

Concept of Pointer



As you can see, **i**’s value is 3 and **j**’s value is **i**’s address

Concept of Pointer

But wait, we can’t use **j** in a program without declaring it

since **j** is a variable that contains the address of **i**, it is declared as,

int *j ;

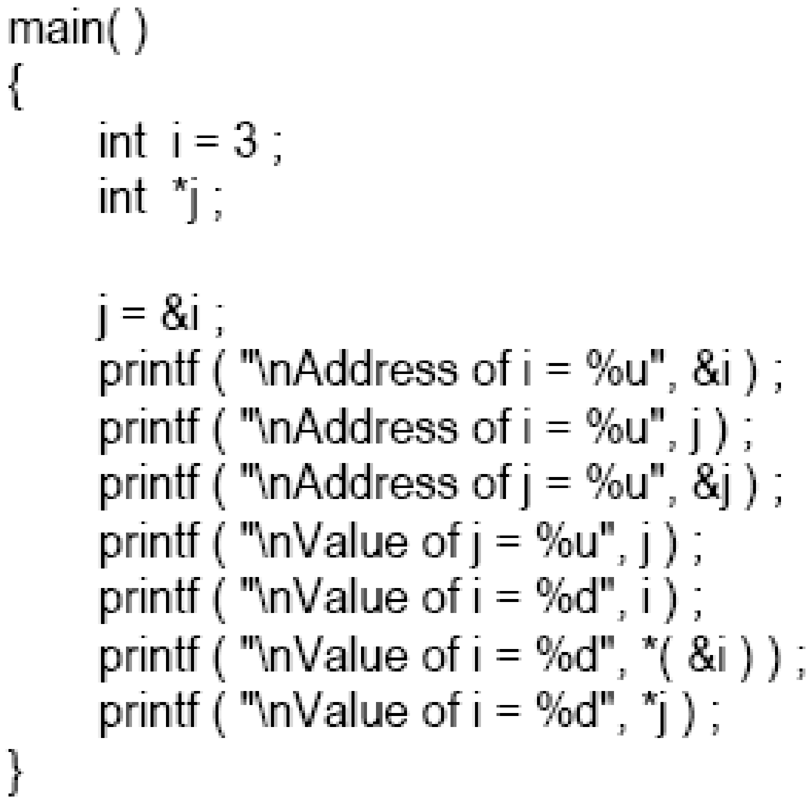
This declaration tells the compiler that **j** will be used to store the address of an integer value

In other words **j** points to an integer

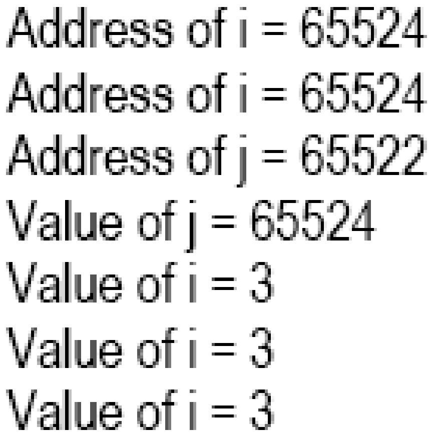
Concept of Pointer

int *j would mean, the value at the address contained in **j** is an **int**

Concept of Pointer



```
main( )
{
        int  i = 3 ;
        int  *j ;

        j = &i ;
        printf ( "\nAddress of i = %u", &i ) ;
        printf ( "\nAddress of i = %u", j ) ;
        printf ( "\nAddress of j = %u", &j ) ;
        printf ( "\nValue of j = %u", j ) ;
        printf ( "\nValue of i = %d", i ) ;
        printf ( "\nValue of i = %d", *( &i ) ) ;
        printf ( "\nValue of i = %d", *j ) ;
}
```



```
Address of i = 65524
Address of i = 65524
Address of j = 65522
Value of j = 65524
Value of i = 3
Value of i = 3
Value of i = 3
```

Concept of Pointer

Look at the following declarations,

1. int *alpha ;
2. char *ch ;
3. float *s ;

The declaration **float *s** does not mean that **s** is going to contain a floating-point value

s is going to contain the address of a floating-point value

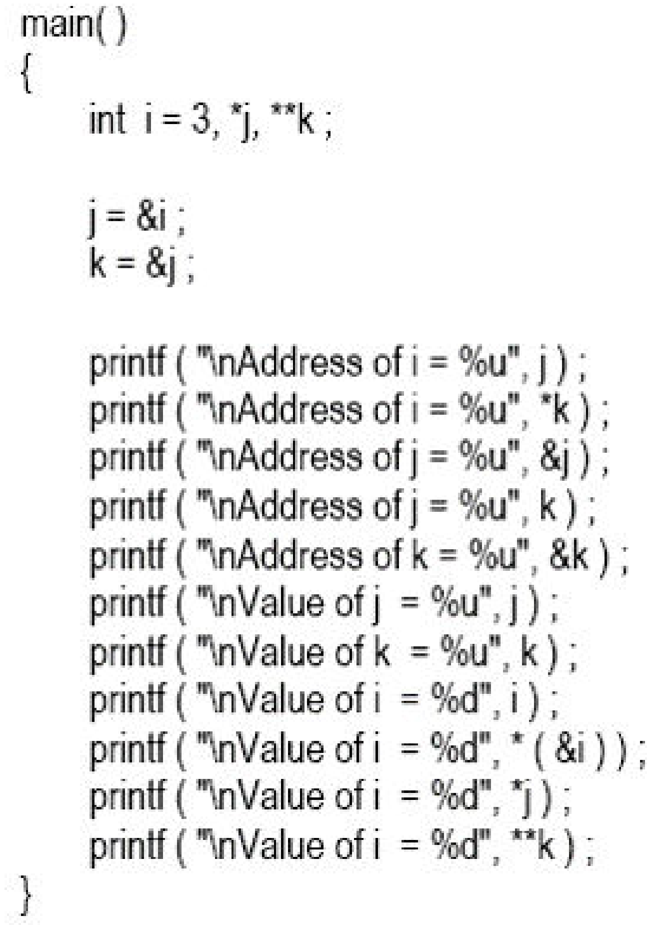
Concept of Pointer

Pointer, we know is a variable that contains address of another variable .

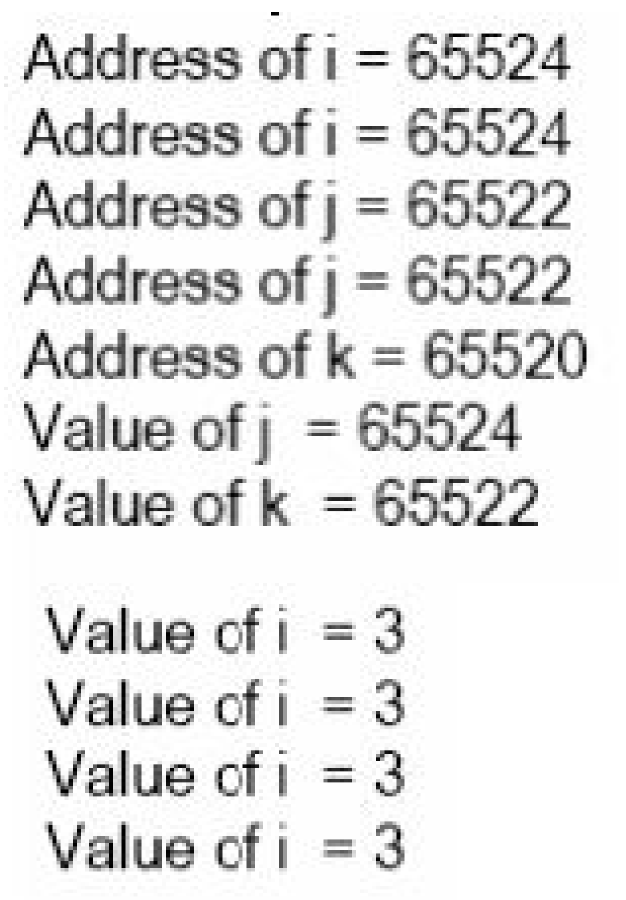
Now this variable itself might be another pointer

Thus, we now have a pointer that contains another pointer’s address

Concept of Pointer

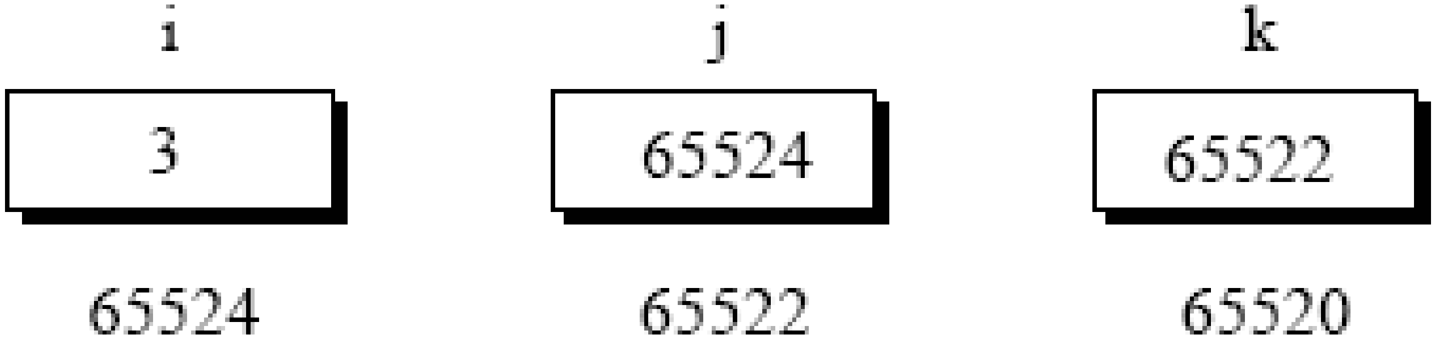


```
main( )
{
      int  i = 3, *j, **k ;

      j = &i ;
      k = &j ;

      printf ( "\nAddress of i = %u", j ) ;
      printf ( "\nAddress of i = %u", *k ) ;
      printf ( "\nAddress of j = %u", &j ) ;
      printf ( "\nAddress of j = %u", k ) ;
      printf ( "\nAddress of k = %u", &k ) ;
      printf ( "\nValue of j  = %u", j ) ;
      printf ( "\nValue of k  = %u", k ) ;
      printf ( "\nValue of i  = %d", i ) ;
      printf ( "\nValue of i  = %d", * ( &i ) ) ;
      printf ( "\nValue of i  = %d", *j ) ;
      printf ( "\nValue of i  = %d", **k ) ;
}
```



```
Address of i = 65524
Address of i = 65524
Address of j = 65522
Address of j = 65522
Address of k = 65520
Value of j  = 65524
Value of k  = 65522

 Value of i  = 3
 Value of i  = 3
 Value of i  = 3
 Value of i  = 3
```

Concept of Pointer



i is an ordinary **int**

j is a pointer to an **int**

k is a pointer to an integer pointer