جامعة بيرزيت
BIRZEIT UNIVERSITY

Pointers & Modular Programming

Computer Science Department

Comp 132

pointer

- **Pointer or pointer variable: A memory cell that stores the address of a data item.**

- **The declaration:**

float *p; // identifies p as a pointer variable of type “pointer to float .”
// This means that we can store the memory address of a type
// float variable in p .

pointer

- **Pointer Type Declaration:**

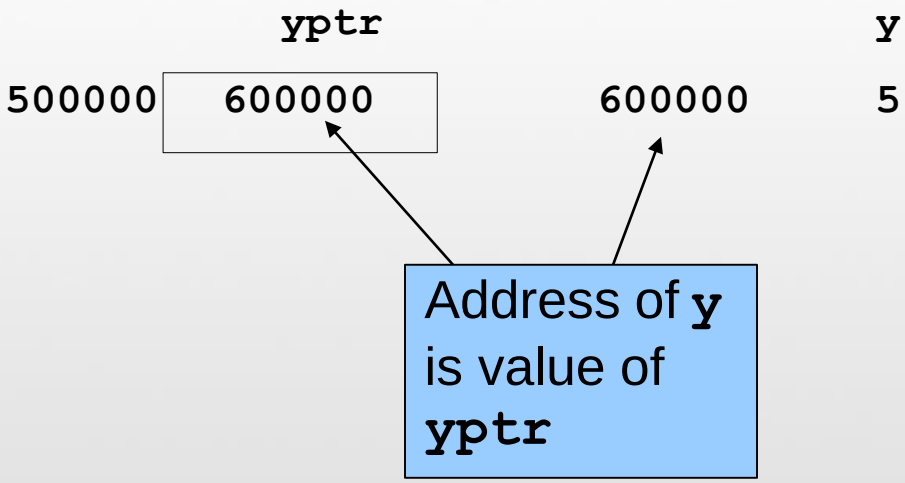
  SYNTAX: ***type * variable ;***
  EXAMPLE: **float *p;**

  The value of the pointer variable p is a memory address

Example (1)

```
int y = 5;
int *yPtr;
yPtr = &y;   //yPtr gets address of y
```

yPtr "points to" y



Example (2)

```
int i = 5;
int *ptr;                          /* declare a pointer variable */
ptr = &i;                          /* store address-of i to ptr */
printf("*ptr = %d\n", *ptr);       /* refer to referee of ptr */
```

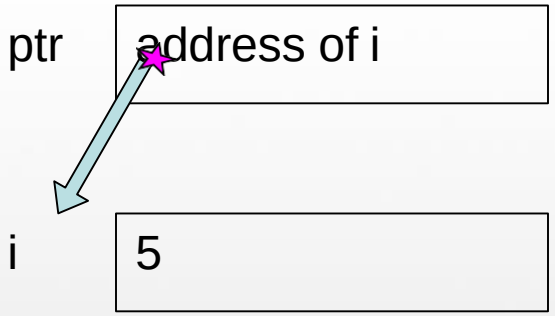
output
*ptr = 5

Example (3):
What actually *ptr* is?

- **ptr** is a variable storing **an address**
- ptr is **NOT** storing the actual value of i

```
Address of i= 0022FF18    Address of ptr=0022FF1C
```

```
int i = 5;
int *ptr;
ptr = &i;
printf("i = %d\n", i);
printf("*ptr = %d\n", *ptr);
printf("ptr = %p\n", ptr);
printf("address of i = %p\n", &i);
printf("address of ptr = %p\n", &ptr);
```



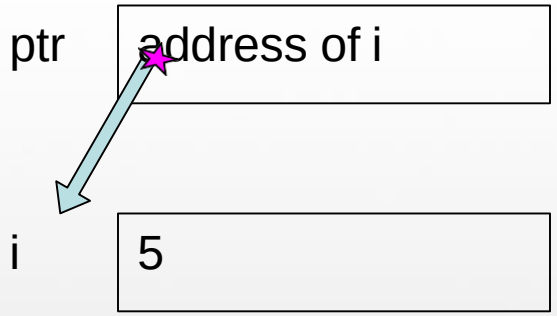
value of ptr = address of i in memory

Example (3):
What actually *ptr* is?

- **ptr** is a variable storing **an address**
- ptr is **NOT** storing the actual value of i

```
Address of i= 0022FF18    Address of ptr=0022FF1C
```

```
int i = 5;
int *ptr;
ptr = &i;
printf("i = %d\n", i);
printf("*ptr = %d\n", *ptr);
printf("ptr = %p\n", ptr);
printf("address of i = %p\n", &i);
printf("address of ptr = %p\n", &ptr);
```



Output:

```
i = 5
*ptr = 5
ptr =0022FF18
address of i = 0022FF18
```

value of ptr = address of i in memory

Example (4)

```c
#include <stdio.h>
int main()
{
    int x, *p;
    p = &x;
    *p = 0;
    printf("x is %d\n", x);
    printf("*p is %d\n", *p);
    *p += 1;
    printf("x is %d\n", x);
    (*p)++;
    printf("x is %d\n", x);
    return 0;
}
```

Example (4)

```c
#include <stdio.h>
int main()
{
    int x, *p;
    p = &x;
    *p = 0;
    printf("x is %d\n", x);
    printf("*p is %d\n", *p);
    *p += 1;
    printf("x is %d\n", x);
    (*p)++;
    printf("x is %d\n", x);
    return 0;
}
```

Output:

```
x is 0
*p is 0
x is 1
x is 2
```

Example (5)

Trace the execution of the following fragment

```
int m = 10, n = 5;
int *mp, *np;
mp = &m;
np = &n;
*mp = *mp + *np;
*np = *mp – *np;
printf("%d %d\n%d %d\n", m, *mp, n, *np);
```

Example (5)

Trace the execution of the following fragment

```
int m = 10, n = 5;
int *mp, *np;
mp = &m;
np = &n;
*mp = *mp + *np;
*np = *mp – *np;
printf("%d %d\n%d %d\n", m, *mp, n, *np);
```

Output:

```
15 15
10 10
```

Examples

```c
#include <stdio.h>
int sum(int,int);
int main()
{
    int num1=4,num2=5;
    int result;
    result=sum(num1,num2);
    printf("The result is %d",result);

    return 0;
}
int sum(int x,int y)
{
    return (x+y);
}
```

```c
#include <stdio.h>
void sum(int*,int,int);
int main()
{
    int num1=4,num2=5;
    int result;
    sum(&result,num1,num2);
    printf("The result is %d",result);

    return 0;
}
void sum(int*res,int x,int y)
{
    *res=x+y;
}
```

Example (6)

Write function to find the sum and the difference between two numbers.

```c
#include <stdio.h>
int sum_difference (int,int,int*);
int main()
{
    int num1,num2,sum,diff;
    printf("Please enter two numbers: ");
    scanf("%d%d",&num1,&num2);
    diff=sum_difference (num1,num2,&sum);
    printf("Sum= %d and difference=%d",sum,diff);
    return 0;
}

int sum_difference (int x,int y,int* sum)
{
    *sum=x+y;
    return (x-y);
}
```

Example (7)

Write a function to :

1. **Find the number of digits in a given number**
2. **Sum of digits**
3. **Reverse a number**

Example:
Please enter a number: 123
number of digits = 3
sum of digits=6
reverse=321

Code

Example (8)

```c
#include<stdio.h>
void interchange(int*,int*);
int main()  {

     int num1,num2;
     printf("Enter num1 and num2: ");
     scanf("%d%d",&num1,&num2);
     interchange(&num1,&num2);

     printf("\nNumber 1 : %d",num1);
     printf("\nNumber 2 : %d",num2);

     return(0);
}
void interchange(int *num1,int *num2)
{
     int temp;
     temp  = *num1;
     *num1 = *num2;
     *num2 = temp;
}
```

Exchanges the values of the two integer variables

Example (9)

Identify and correct the errors in the following code fragment, given the correct output (%p is used to print a pointer):

```
int y = 3;
int *yptr;
yptr = &y;
printf("The value of y is %d\n", *yptr);
printf("The address of y is %p\n",*yptr);
```

Change "*yptr" in the above statement to "yptr" or "&y"

Output:

The value of y is 3

The address of y is 2063865468

Example (10): Output

```c
#include <stdio.h>
int main()
{
    int i = 0, j = 5;
    int *y;
    y=&j;
    for( i = 0; i <= 4; i++ )
    {
        *y  = *y + i;
    }
    printf( "The final value of j is %d.\n", j );
    return 0;
}
```

Example (10): Output

```c
#include <stdio.h>
int main()
{
    int i = 0, j = 5;
    int *y;
    y=&j;
    for( i = 0; i <= 4; i++ )
    {
        *y  = *y + i;
    }
    printf( "The final value of j is %d.\n", j );
    return 0;
}
```

The final value of j is 15.

Example (11)

C program to find square and cube of given number

```c
#include <stdio.h>
int square_cube(int,int*);
int main()
{
    int num,square,cube;
    printf("Please enter a number : ");
    scanf("%d",&num);
    square=square_cube(num,&cube);
    printf("square=%d\ncube=%d",square,cube);
    return 0;
}
int square_cube(int num,int*cube)
{
   int square;
   square=num*num;
   *cube=num*num*num;
   return square;
}
```

```
Please enter a number : 2
square=4
cube=8
```

Example (12)

```c
#include <stdio.h>
int x=2;
void fun1(int,int*,int);
void fun2(int,int*);
int main()
{
    int num1=2,num2=3,res=0;
    x=num1+1;
    printf("num1=%d num2=%d  res=%d  x=%d\n",num1,num2,res,x);
    fun1 (num1,&res,num2);
    printf("num1=%d num2=%d  res=%d  x=%d\n",num1,num2,res,x);
    fun2(num2,&res);
    printf("num1=%d num2=%d  res=%d  x=%d\n",num1,num2,res,x);
    return 0;
}
void fun1(int x,int* y,int z)
{
    *y=x+z;
    *y=x+2;

}
void fun2(int y,int* z)
{
    *z=x+2;
    *z=y+3;
    x++;
}
```

Example (12)

```c
#include <stdio.h>
int x=2;
void fun1(int,int*,int);
void fun2(int,int*);
int main()
{
    int num1=2,num2=3,res=0;
    x=num1+1;
    printf("num1=%d num2=%d  res=%d  x=%d\n",num1,num2,res,x);
    fun1 (num1,&res,num2);
    printf("num1=%d num2=%d  res=%d  x=%d\n",num1,num2,res,x);
    fun2(num2,&res);
    printf("num1=%d num2=%d  res=%d  x=%d\n",num1,num2,res,x);
    return 0;
}
void fun1(int x,int* y,int z)
{
    *y=x+z;
    *y=x+2;

}
void fun2(int y,int* z)
{
    *z=x+2;
    *z=y+3;
    x++;
}
```

code

Output:

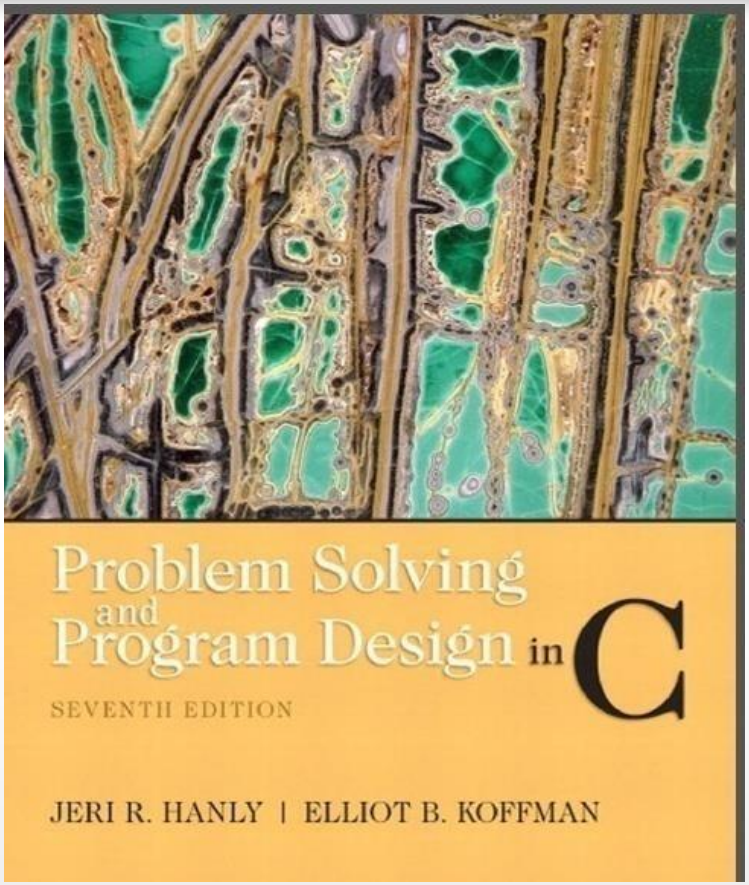
num1=2 num2=3 res=0 x=3

num1=2 num2=3 res=4 x=3

num1=2 num2=3 res=6 x=4

Question?

“Success is the sum of small efforts, repeated day in and day out.”
Robert Collier



References:

Problem Solving & Program Design in C (main reference)

http://www.programmingsimplified.com/c-program-print-stars-pyramid