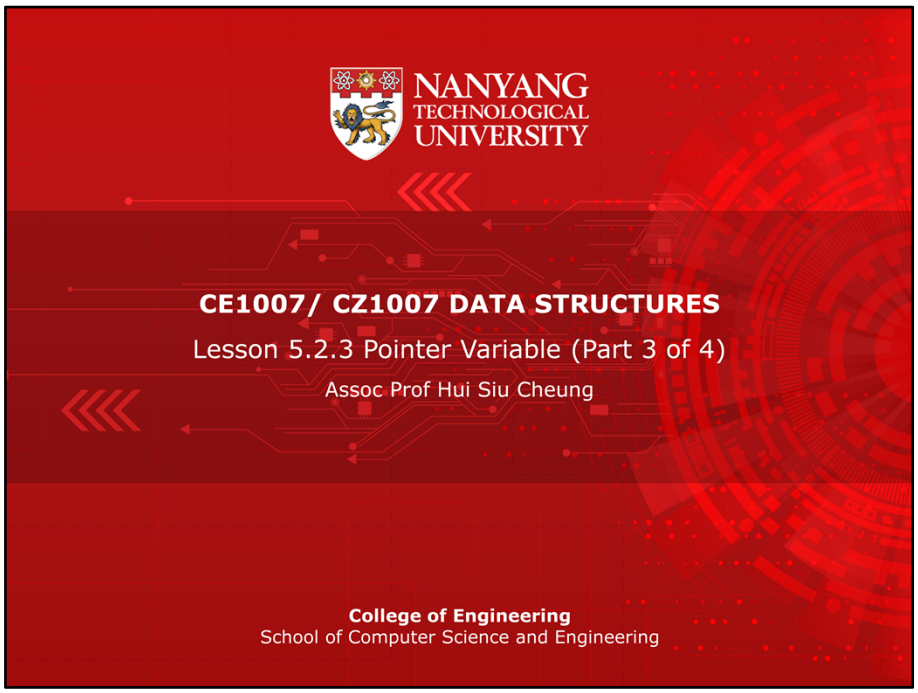
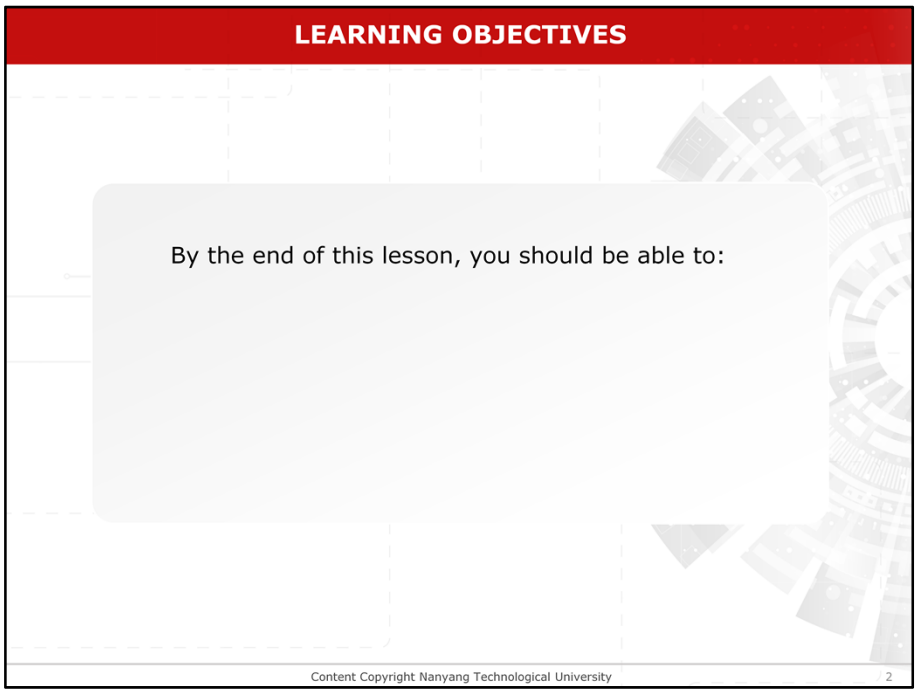
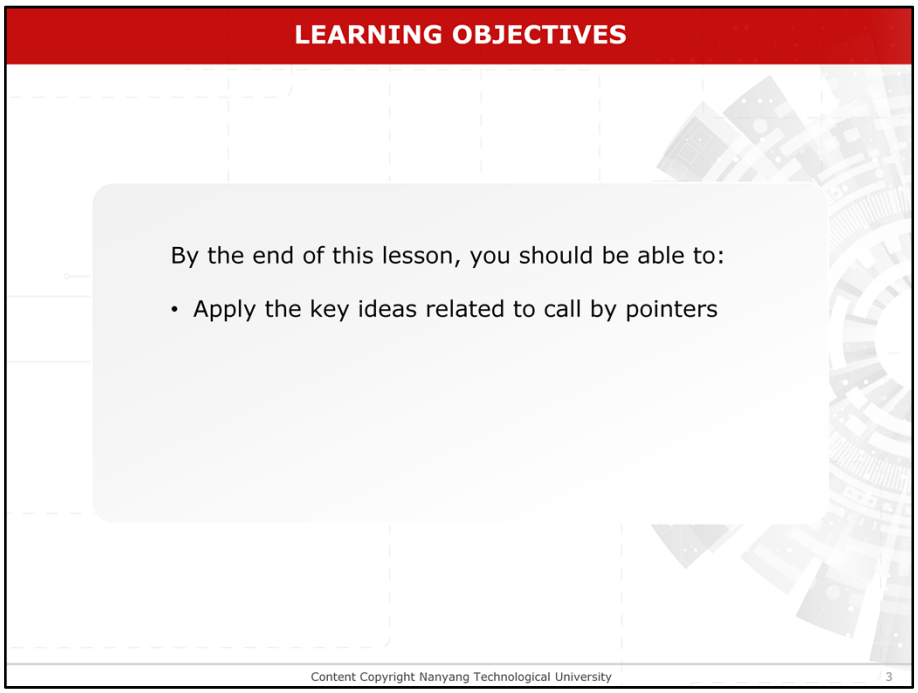


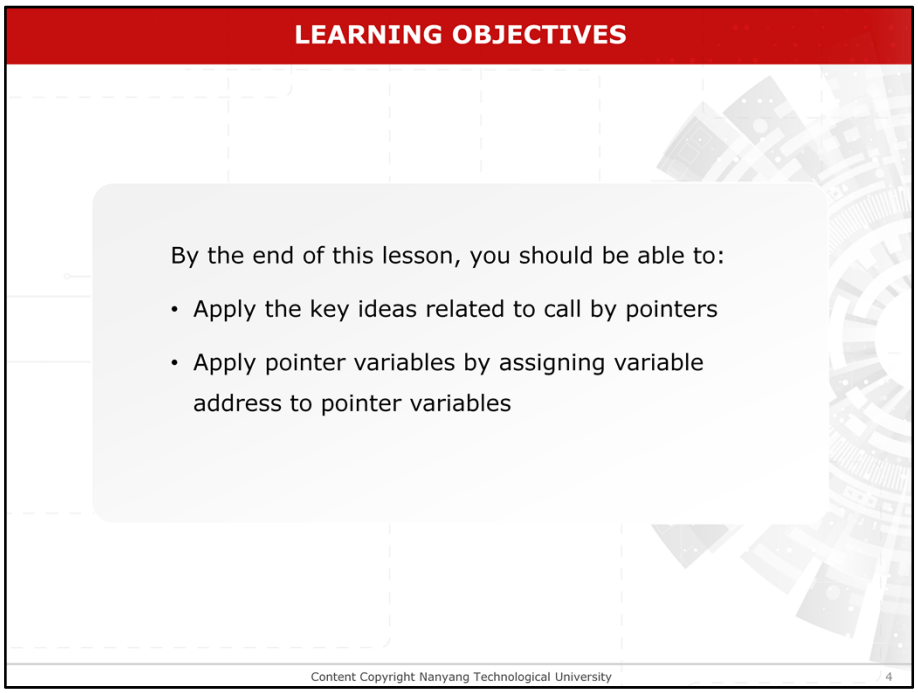
This lesson will show you how to further apply what you have learnt in the previous lesson using a programming code example with detailed explanation.



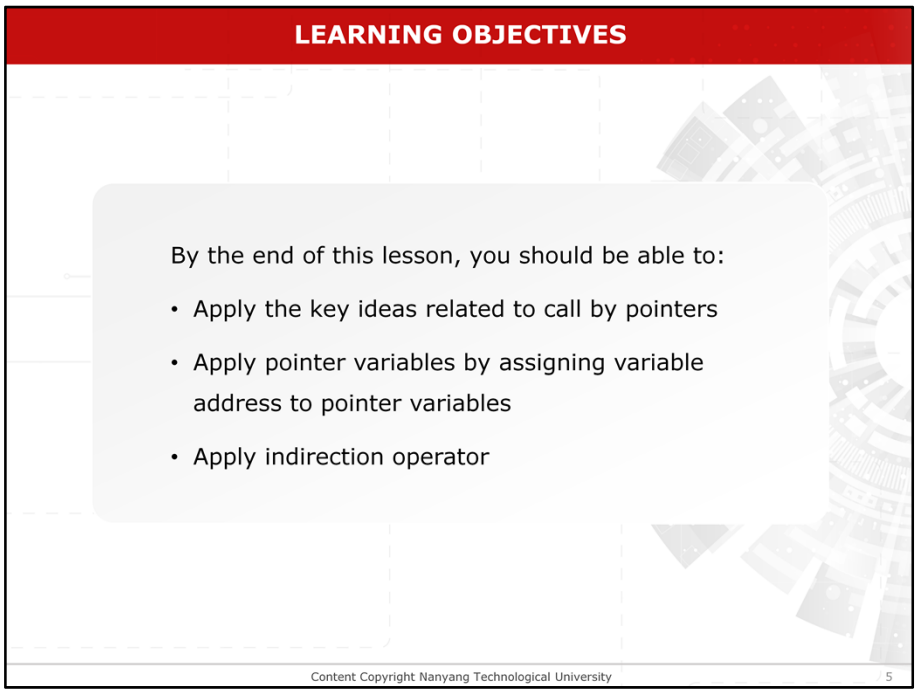
By the end of this lesson, you should be able to:



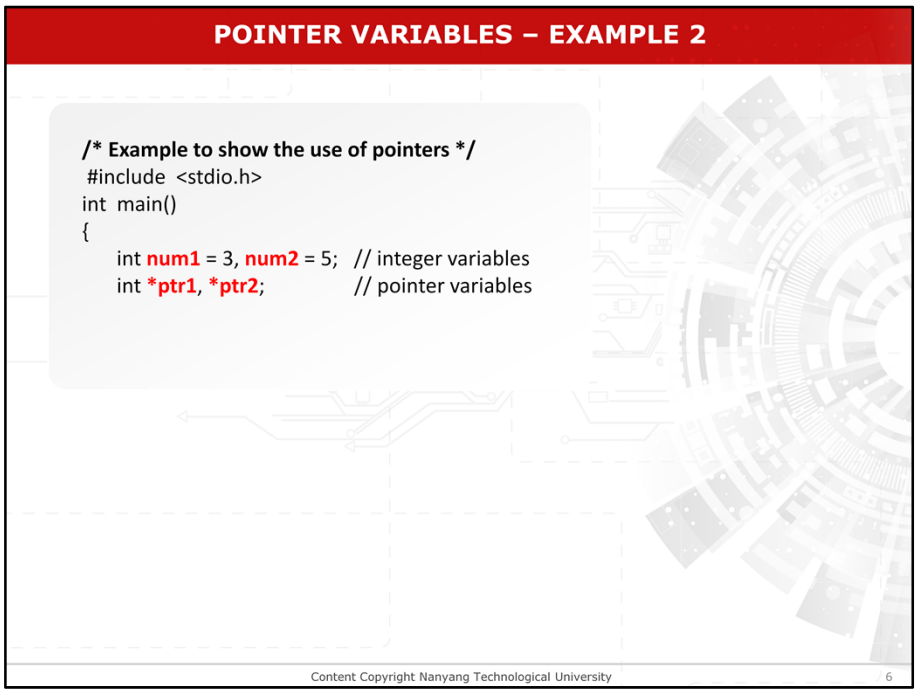
Apply the key ideas related to call by pointers



Apply pointer variables by assigning variable address to pointer variables

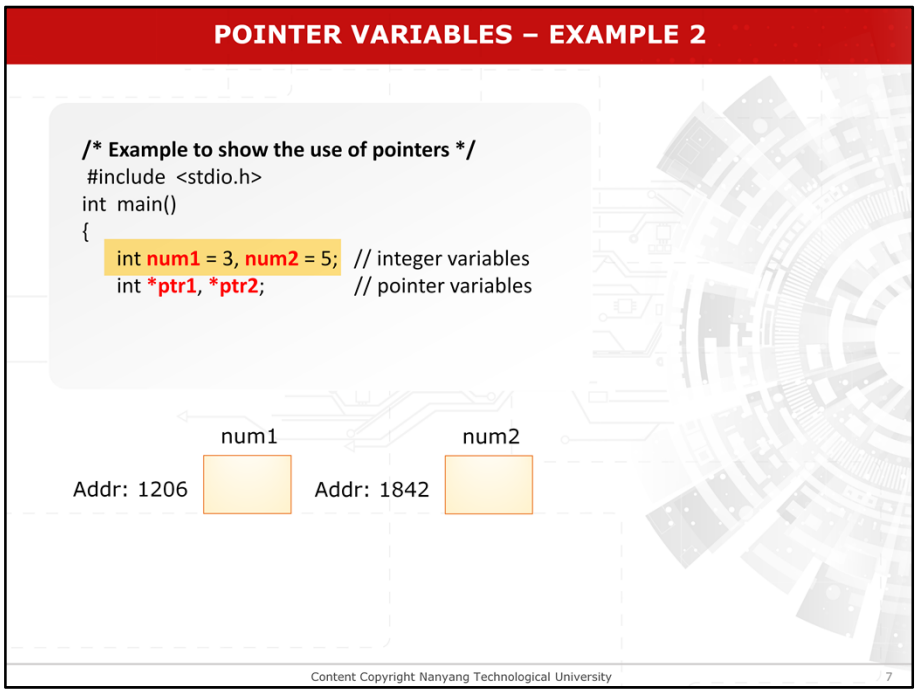


Apply indirection operator

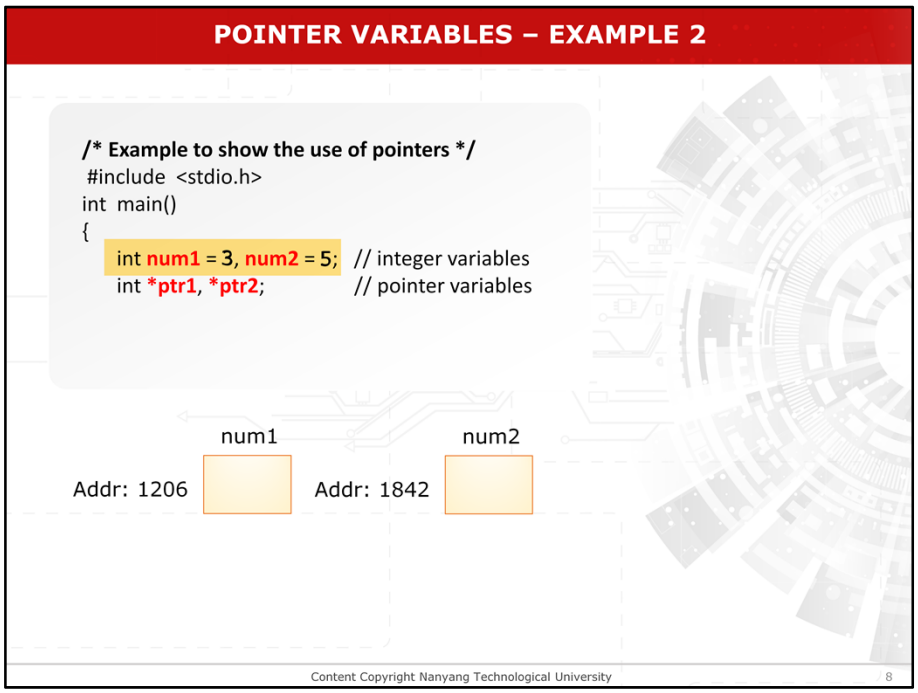


This is a C program that uses the concepts of pointers. Let's go through the code in detail, part by parts.

This first part of the code is very similar to what you learnt in previous lesson. So we will go through briefly for this part.



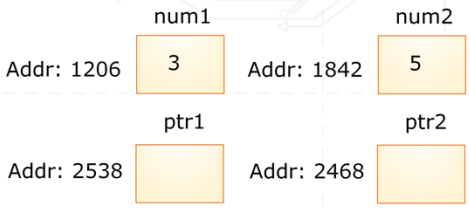
This statement declares integer data type for variable num1 and num2. Let's assume the system allocated memory address 1206 to variable num1 and address 1842 to num2.



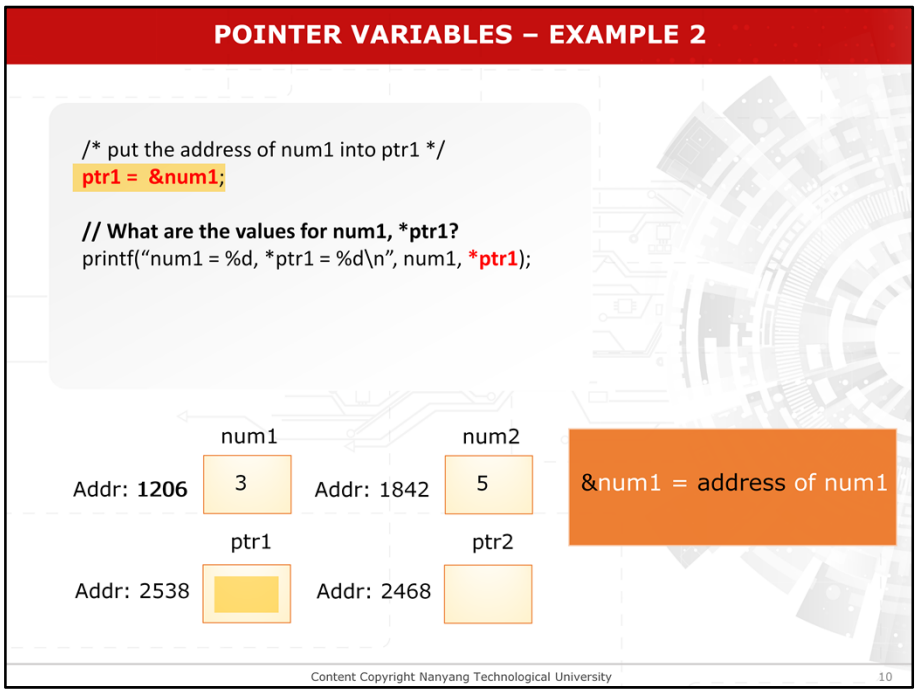
And both variable num1 and num2 have been initialized with the value 3 and 5 respectively.

POINTER VARIABLES – EXAMPLE 2

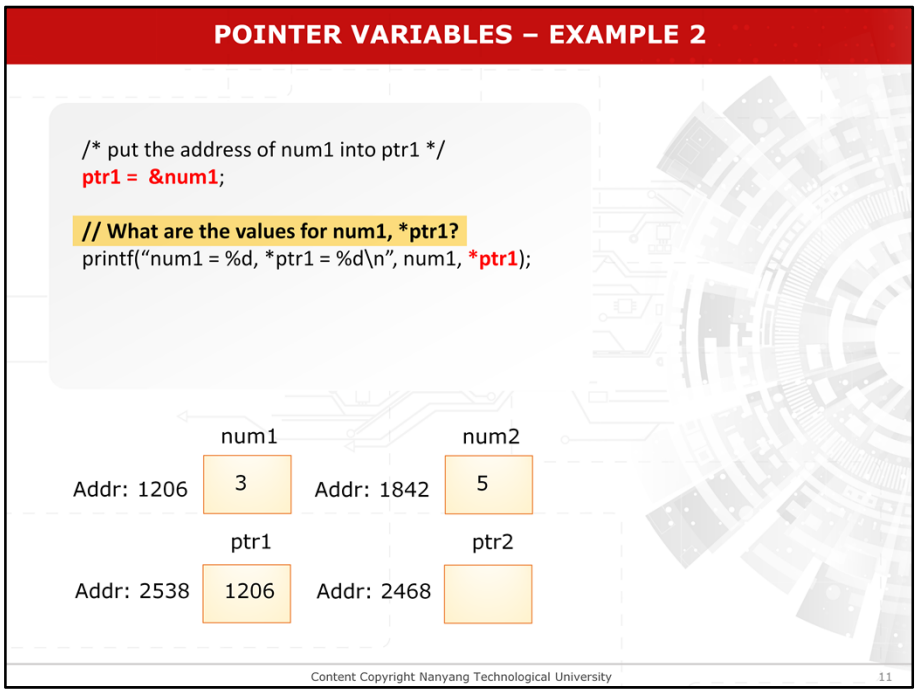
```
/* Example to show the use of pointers */
 #include <stdio.h>
int  main()
{
     int num1 = 3, num2 = 5;   // integer variables
     int *ptr1, *ptr2;         // pointer variables
```



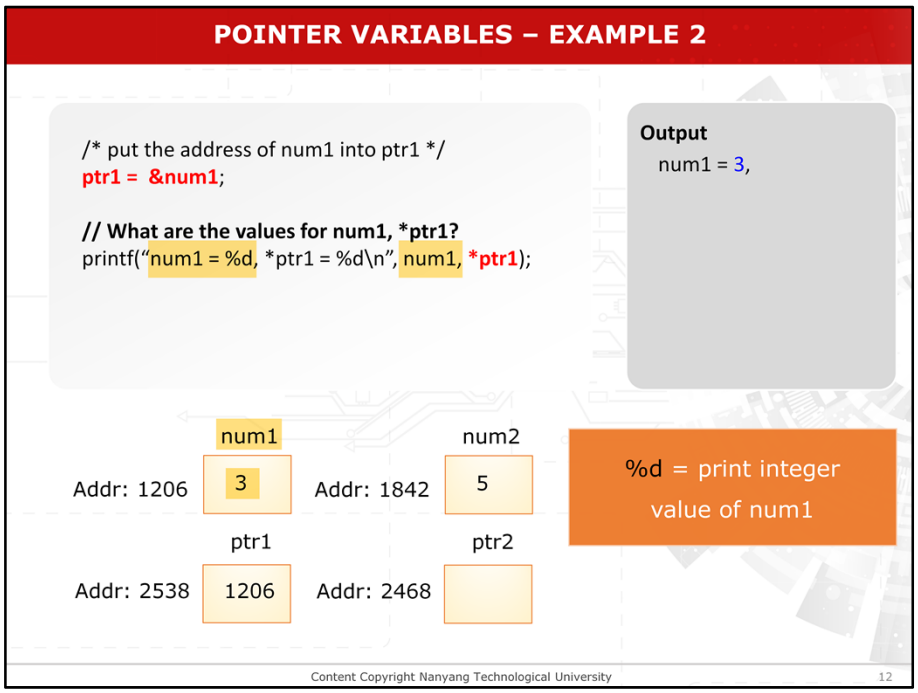
In the highlighted statement, pointer variables namely pointer 1 and pointer 2 are declared so the system allocated some memory address to these variables. But there is no value stored in theses pointer variables yet.



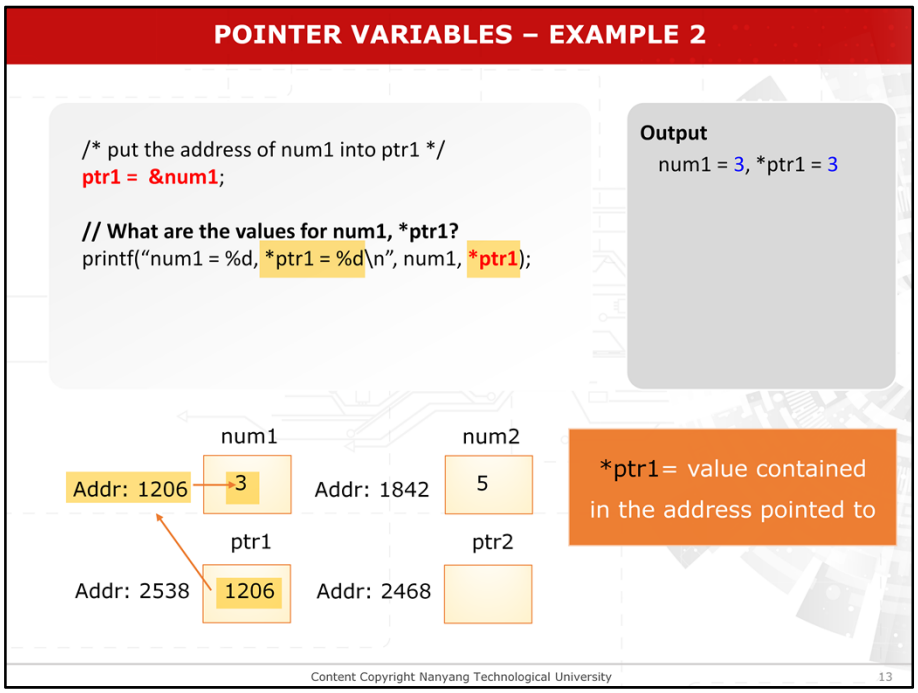
This statement pointer1 equals ampersand num1 assigns the address of num1 to pointer 1



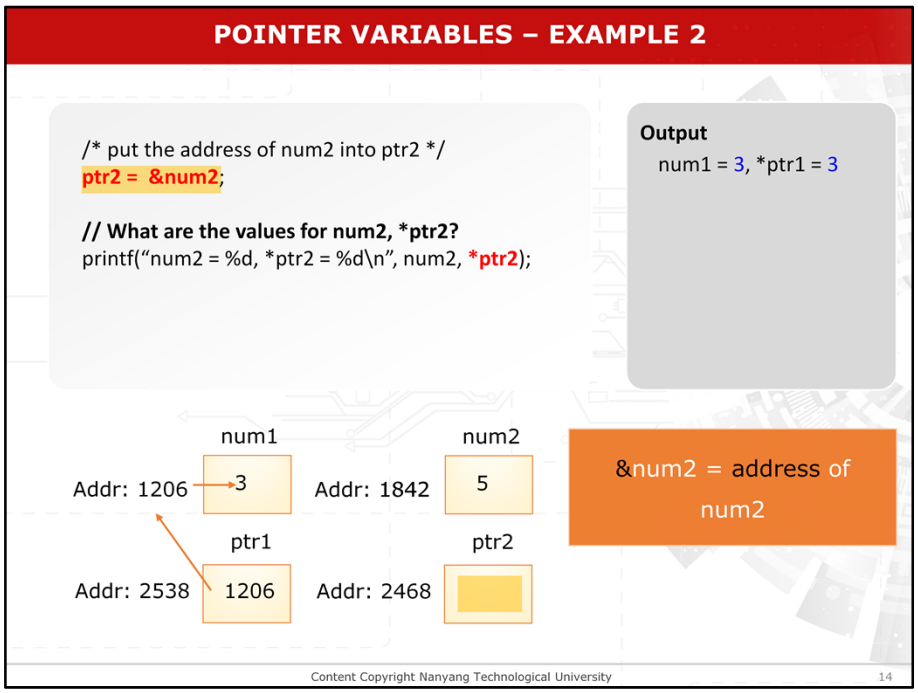
What do you think are the values for num1 and asterisk pointer1?



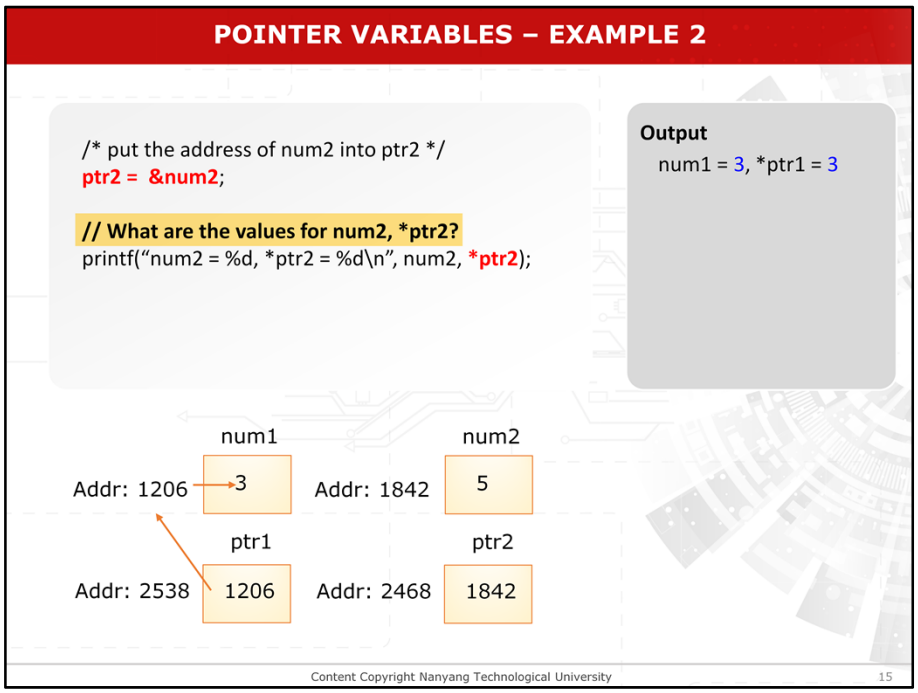
The num1 equals percent d in the printf statement will gives an output of num1 equals 3



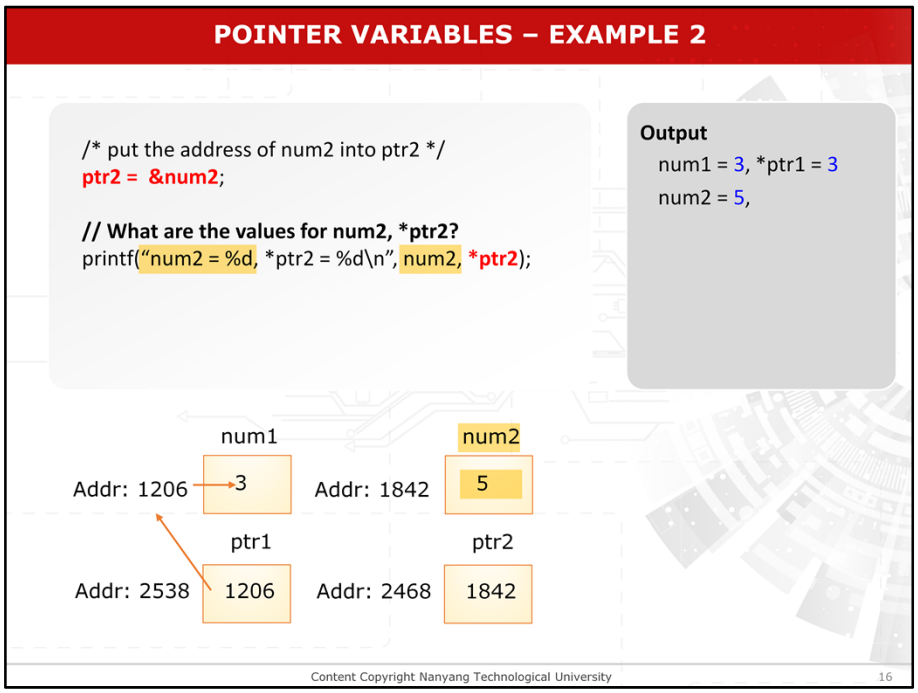
The part on asterisk pointer 1 equals percent d in the printf statement will gives an output of asterisk pointer 1 equals 3. Pointer 1 points to the address of num1 and asterisk pointer 1 retrieve the value contains in num1



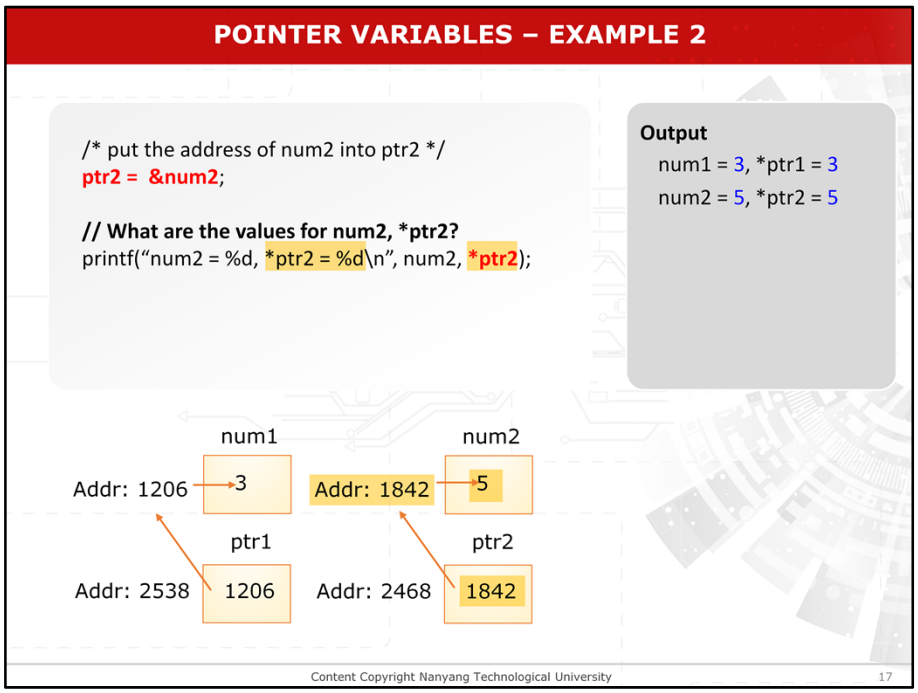
Pointer 2 is assigned the address of num2.



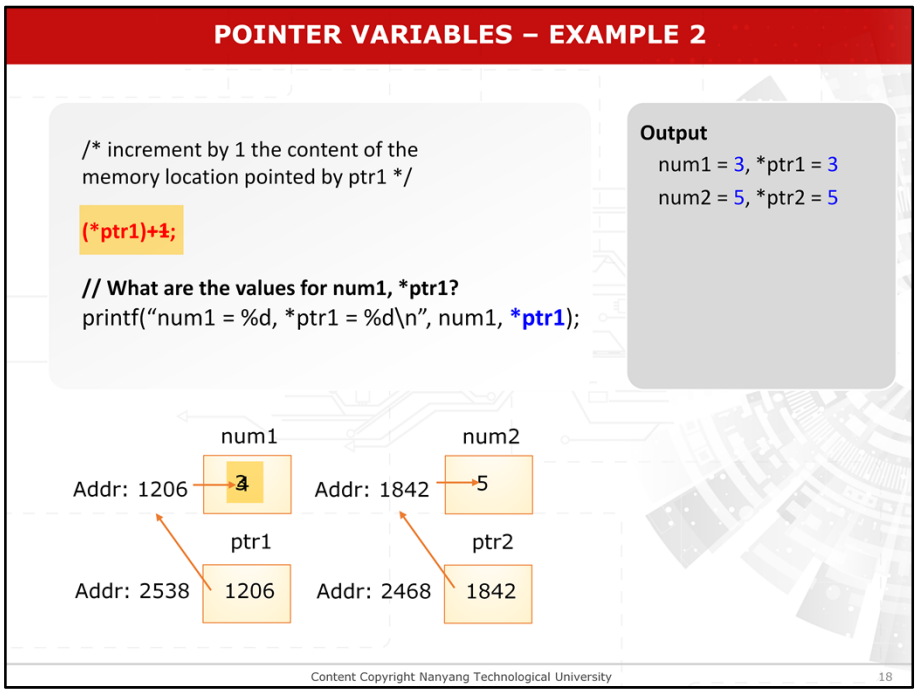
What do you think are the values for num2 and asterisk pointer 2?



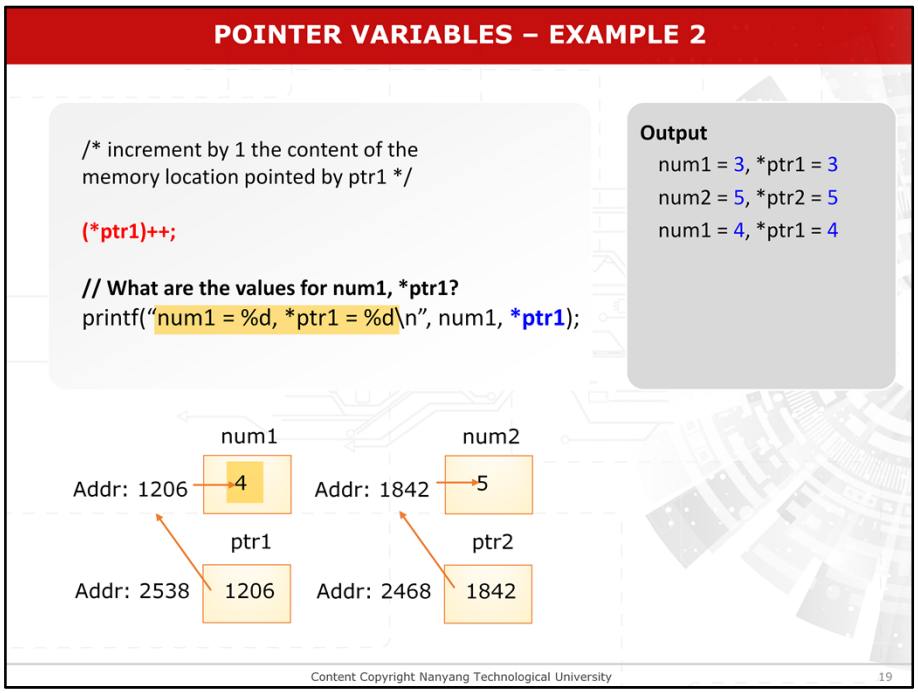
The output should give num2 equals 5



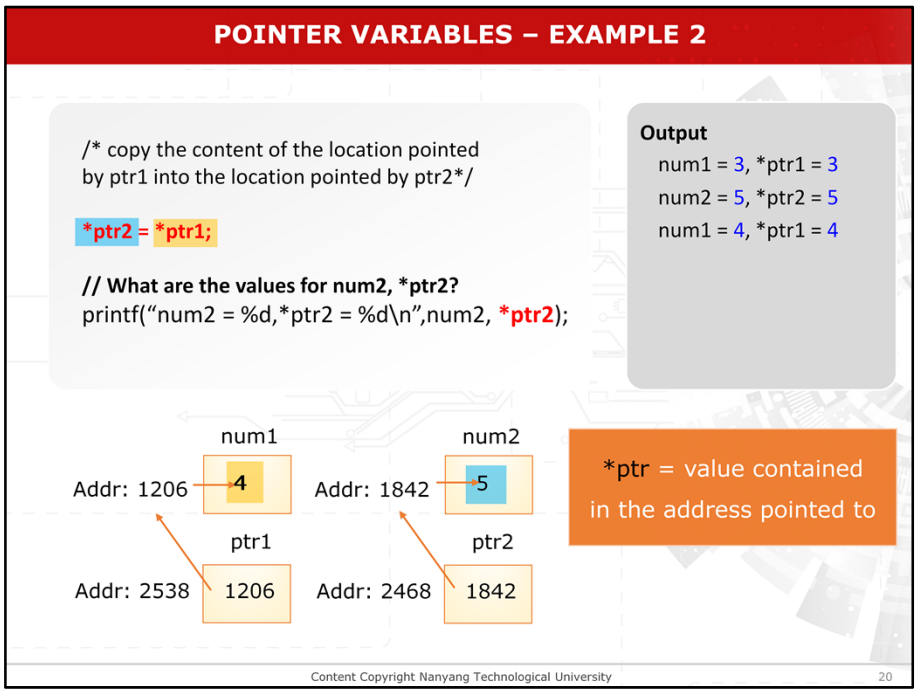
And asterisk pointer 2 equals 5.



The highlighted statement asterisk pointer 1 plus plus increments 1 to the content of the memory location pointed to by **pointer 1**



Therefore, the output shows num1 equals 4 and asterisk pointer 1 equals 4 as well.



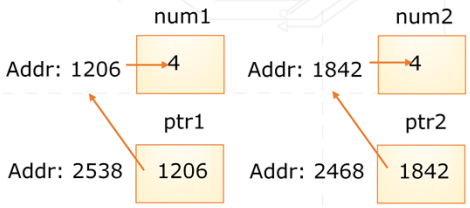
In this highlighted statement of asterisk pointer 2 equals asterisk pointer 1, what will happen is that the content of the location pointed by pointer 1 is copied into the location pointed by pointer 2.

POINTER VARIABLES – EXAMPLE 2

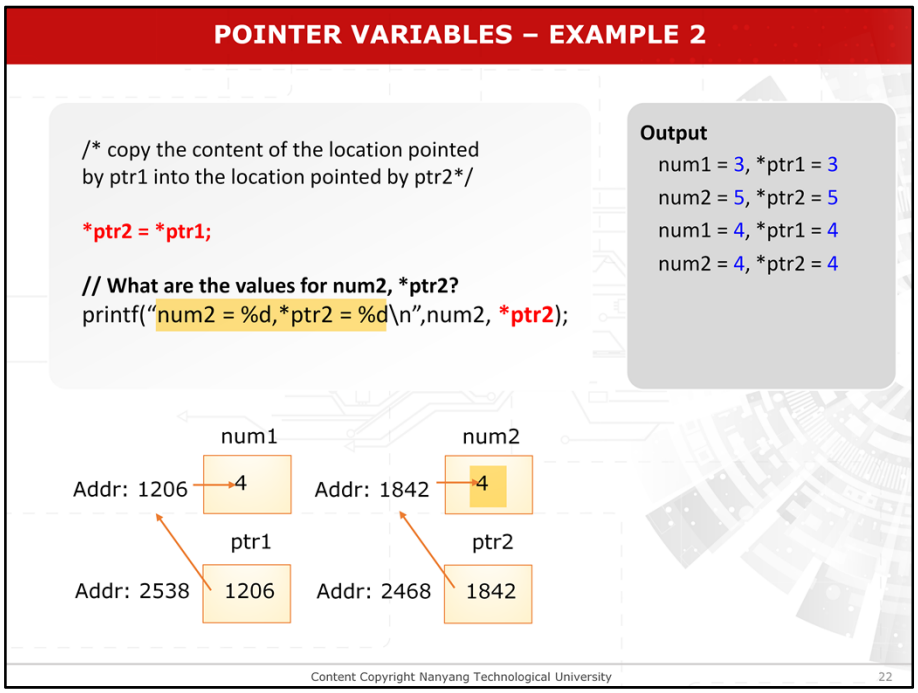
```
/* copy the content of the location pointed
by ptr1 into the location pointed by ptr2*/

*ptr2 = *ptr1;

// What are the values for num2, *ptr2?
printf("num2 = %d,*ptr2 = %d\n",num2, *ptr2);
```

Output

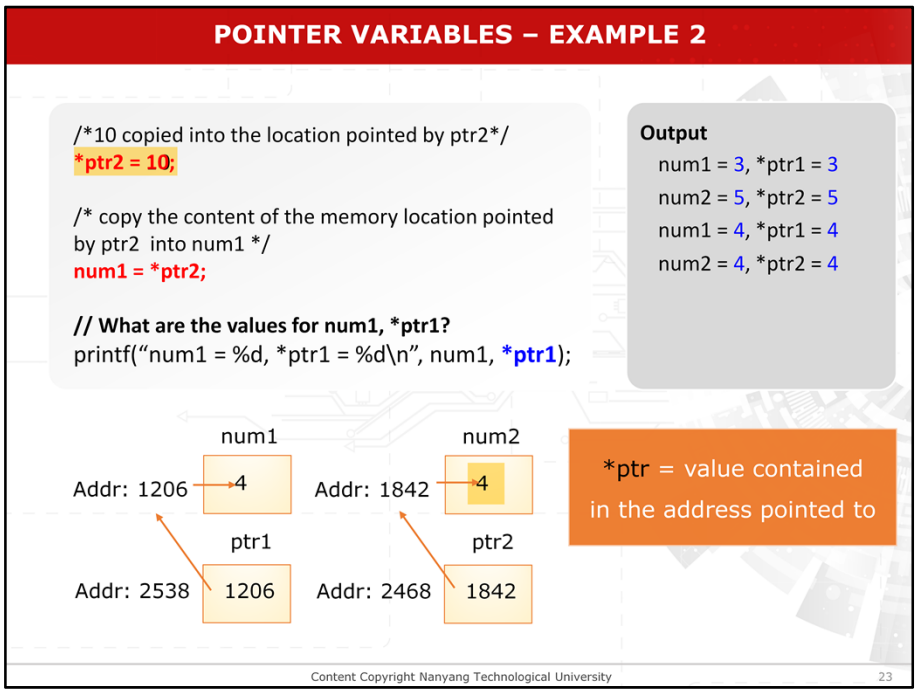
num1 = 3, *ptr1 = 3
num2 = 5, *ptr2 = 5
num1 = 4, *ptr1 = 4



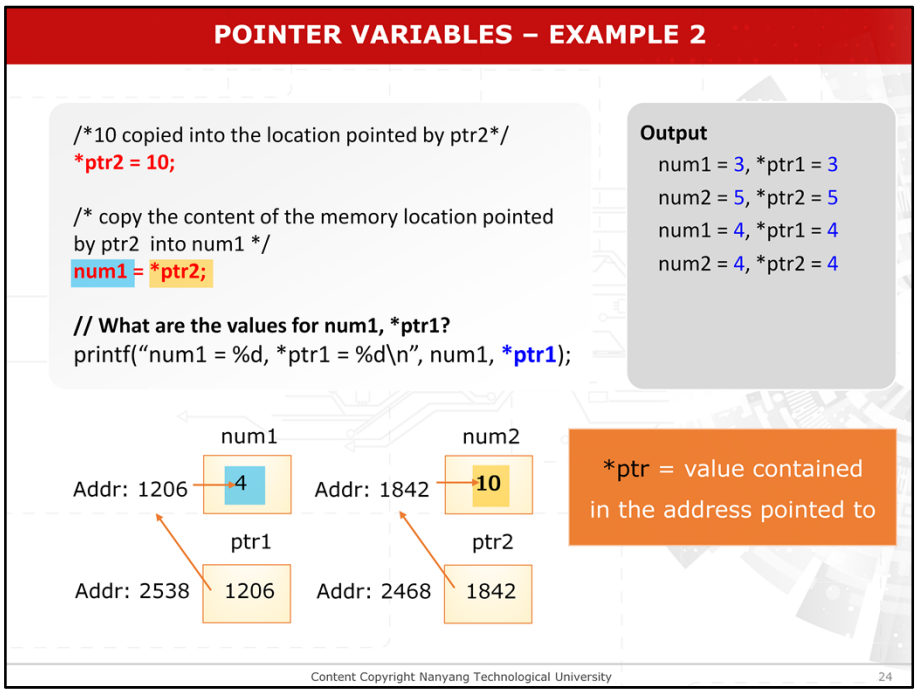
What do you think are the values of num2 and asterisk pointer 2?



So the output should be num2 equals 4, and asterisk pointer 2 equals 4 as well.



In the highlighted statement of asterisk pointer 2 equals 10, the value 10 will be copied into the location pointed by pointer 2.



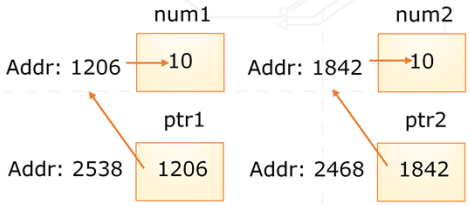
In the highlighted statement of num 1 equals asterisk pointer 2 will copy the content of the memory location pointed by pointer 2 into num1

POINTER VARIABLES – EXAMPLE 2

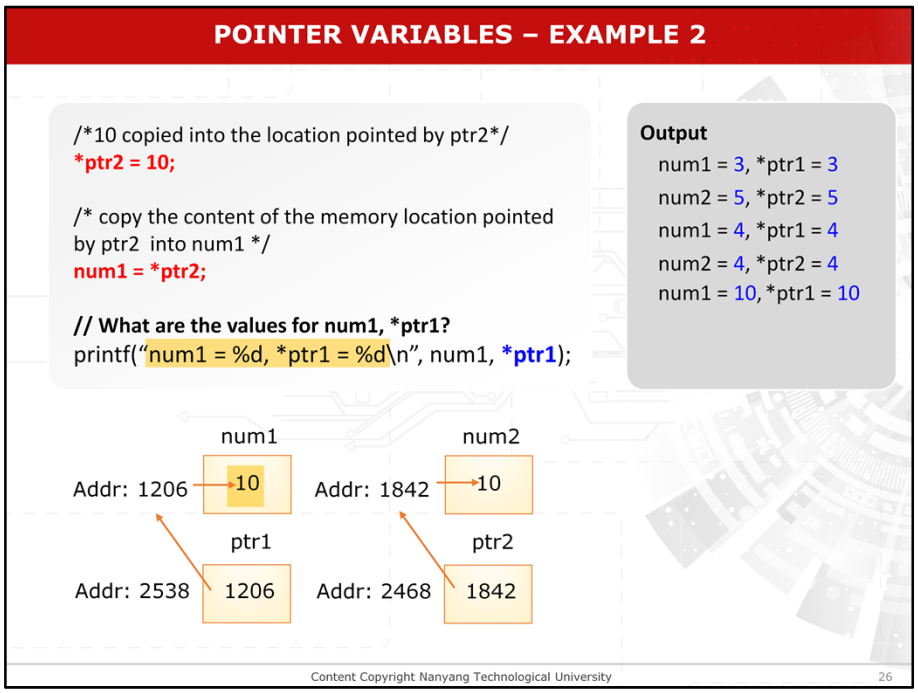
```
/*10 copied into the location pointed by ptr2*/
*ptr2 = 10;

/* copy the content of the memory location pointed
by ptr2  into num1 */
num1 = *ptr2;

// What are the values for num1, *ptr1?
printf("num1 = %d, *ptr1 = %d\n", num1, *ptr1);
```

Output

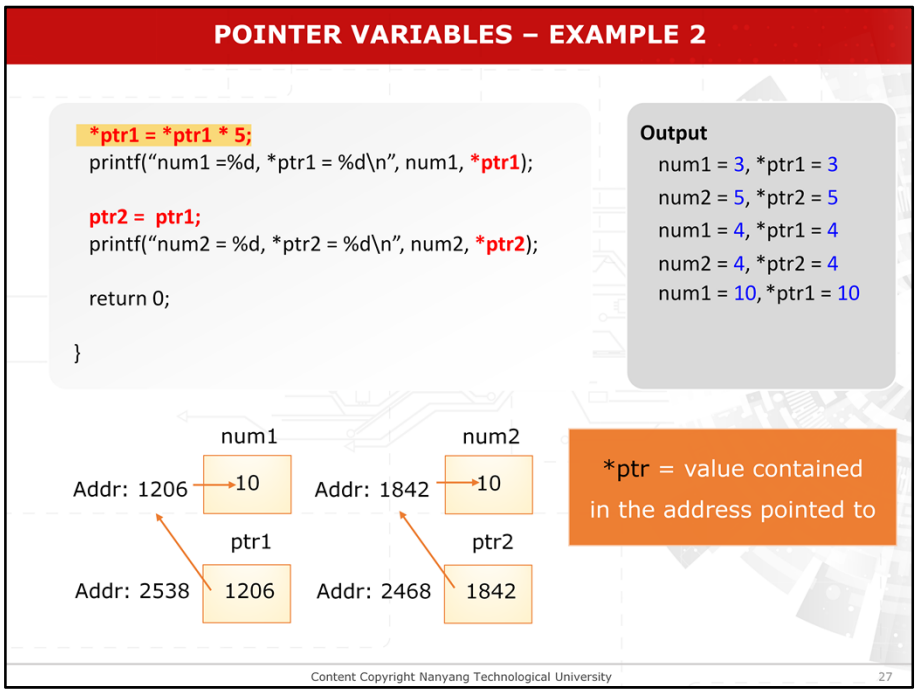
num1 = 3, *ptr1 = 3
num2 = 5, *ptr2 = 5
num1 = 4, *ptr1 = 4
num2 = 4, *ptr2 = 4



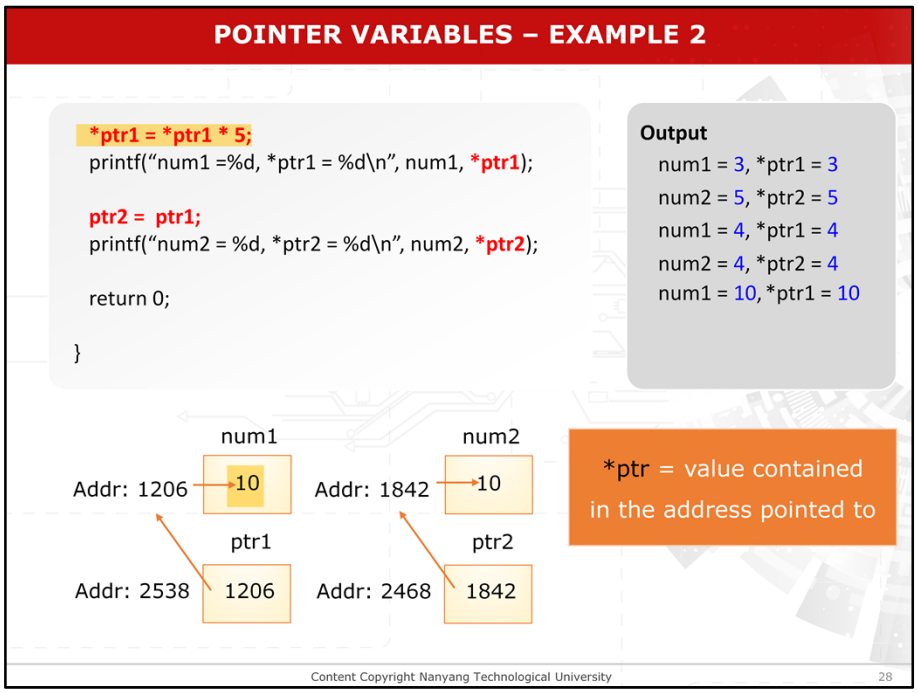
What do you think are the values for num1 and asterisk pointer 1?



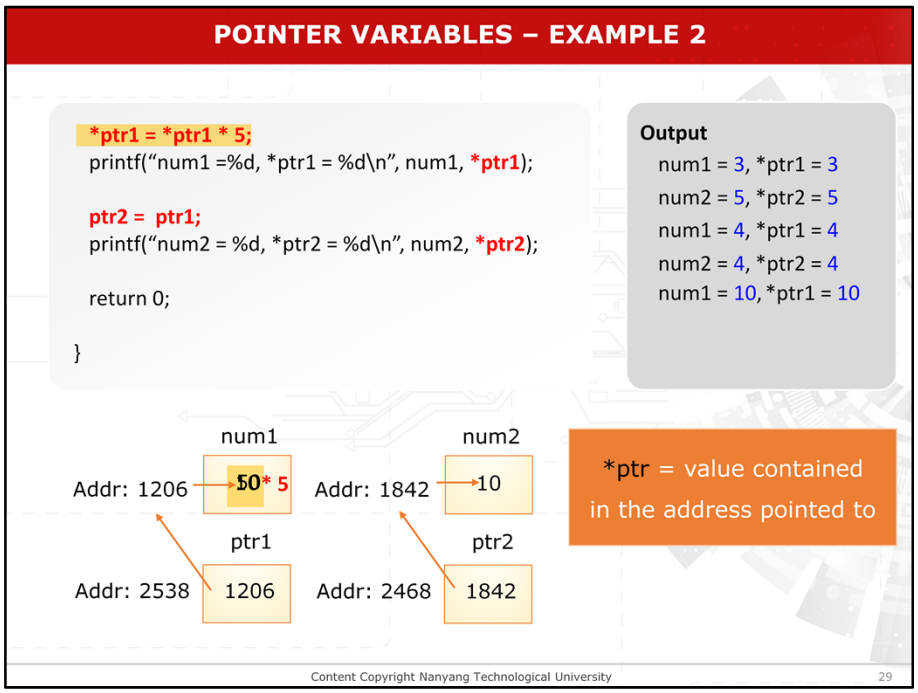
The output will show num1 equals 10 and asterisk pointer 1 equals 10.



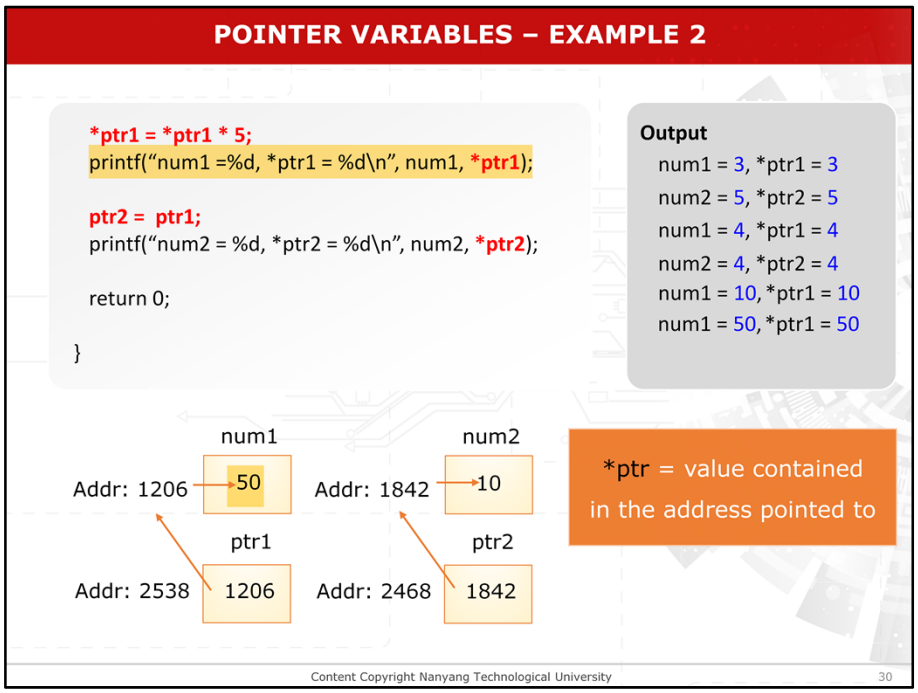
The highlighted statement is asterisk pointer 1 equals asterisk pointer 1 multiply by the value of 5.



Since asterisk pointer 1 equals 10, we have 10 multiply by 5 which gives the value of 50.



The new value 50 is assigned to the content of the memory location pointed to by **pointer 1.**



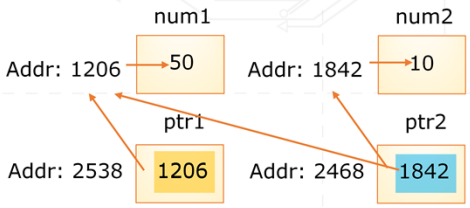
This print f statement will give an output of num1 equals 50 and asterisk pointer 1 equals 50.

POINTER VARIABLES – EXAMPLE 2

```
    *ptr1 = *ptr1 * 5;
    printf("num1 =%d, *ptr1 = %d\n", num1, *ptr1);

    ptr2 = ptr1;
    printf("num2 = %d, *ptr2 = %d\n", num2, *ptr2);

    return 0;

}
```

Output

num1 = 3, *ptr1 = 3
num2 = 5, *ptr2 = 5
num1 = 4, *ptr1 = 4
num2 = 4, *ptr2 = 4
num1 = 10, *ptr1 = 10
num1 = 50, *ptr1 = 50



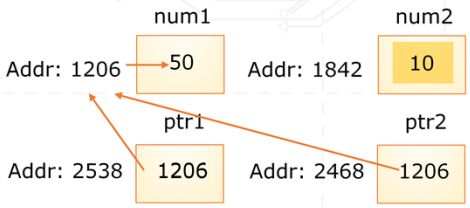
In the highlighted statement where pointer 2 equals pointer 1, it copies the address in **pointer 1** into **pointer2**, so that the pointer variable **pointer2** points to the same memory location as **pointer1**.

POINTER VARIABLES – EXAMPLE 2

```
*ptr1 = *ptr1 * 5;
printf("num1 =%d, *ptr1 = %d\n", num1, *ptr1);

ptr2 =  ptr1;
printf("num2 = %d, *ptr2 = %d\n", num2, *ptr2);

return 0;

}
```

Output

num1 = 3, *ptr1 = 3
num2 = 5, *ptr2 = 5
num1 = 4, *ptr1 = 4
num2 = 4, *ptr2 = 4
num1 = 10, *ptr1 = 10
num1 = 50, *ptr1 = 50
num2 = 10,



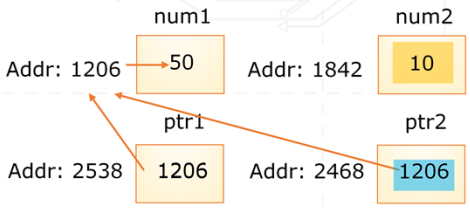
So in the print f statement, num2 equals 10.

POINTER VARIABLES – EXAMPLE 2

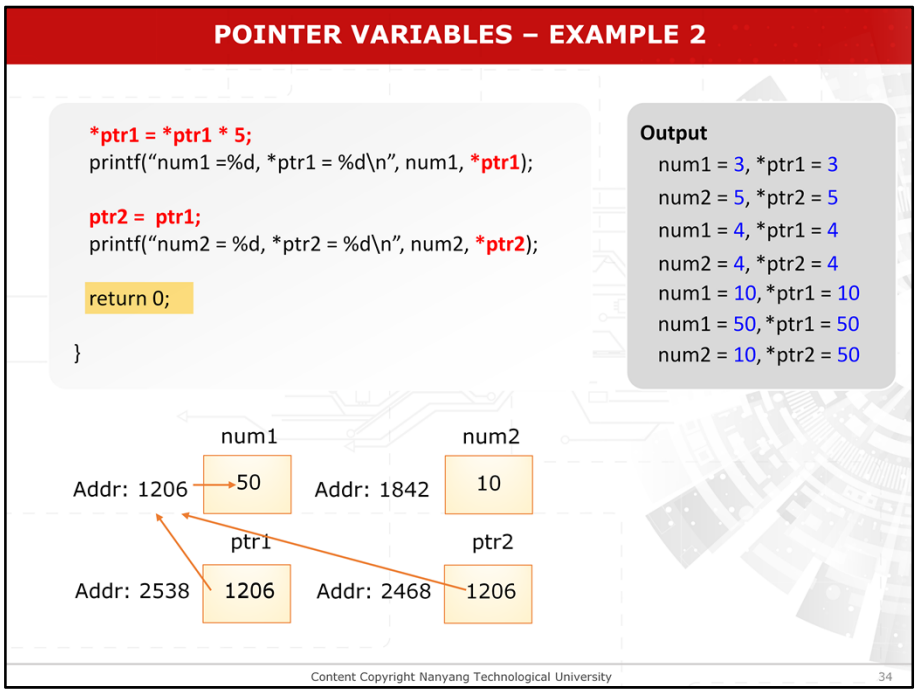
```
    *ptr1 = *ptr1 * 5;
    printf("num1 =%d, *ptr1 = %d\n", num1, *ptr1);

    ptr2 =  ptr1;
    printf("num2 = %d, *ptr2 = %d\n", num2, *ptr2);

    return 0;

}
```

Output

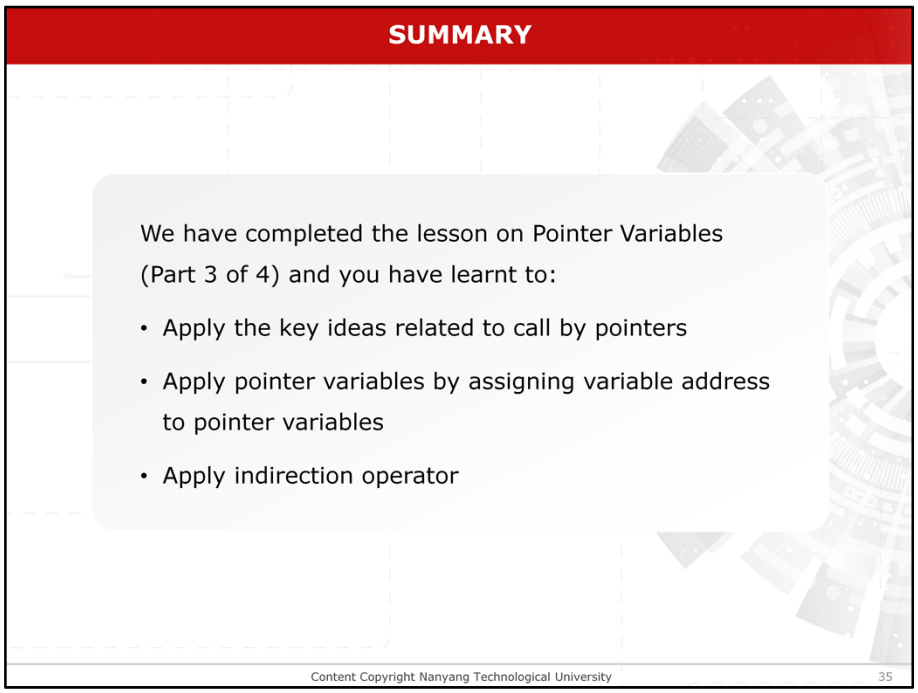
num1 = 3, *ptr1 = 3
num2 = 5, *ptr2 = 5
num1 = 4, *ptr1 = 4
num2 = 4, *ptr2 = 4
num1 = 10, *ptr1 = 10
num1 = 50, *ptr1 = 50
num2 = 10, *ptr2 = 50



And asterisk pointer 2 equals 50.



Return 0 denotes the end of the program.



Through this code example, you have learnt the points listed. In the next lesson, we will summarise the main learning points on Pointers Variables.