



C Pointers



THE

C

PROGRAMMING
LANGUAGE





C Pointers



Pointer to Pointer





C Pointers



We already know that a pointer holds the address of another variable of same type. When a pointer holds the address of another pointer then such type of pointer is known as pointer-to-pointer or double pointer. In this lesson, we will learn what is a double pointer, how to declare them and how to use them in C programming





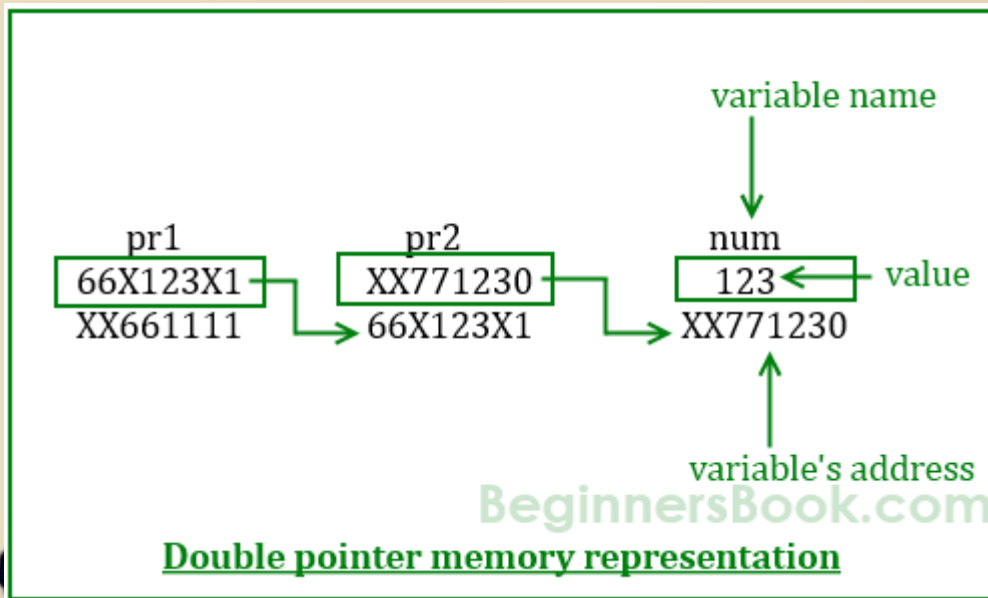
C Pointers



How to declare a Pointer to Pointer (Double Pointer) in C?

```
int **pr;
```

Here pr is a double pointer. There must be two *'s in the declaration of double pointer.



As per the diagram, pr2 is a normal pointer that holds the address of an integer variable num. There is another pointer pr1 in the diagram that holds the address of another pointer pr2, the pointer pr1 here is a pointer-to-pointer (or double pointer).



C Pointers



Example of double Pointer

Lets write a C program based on the diagram that we have seen above.

```
p2p_1.c x
1  #include <stdio.h>
2  int main()
3  {
4      int num=123;
5
6      //A normal pointer pr2
7      int *pr2;
8
9      //This pointer pr2 is a double pointer
10     int **pr1;
11
12     /* Assigning the address of variable num to the
13      * pointer pr2
14      */
15     pr2 = &num;
16
17     /* Assigning the address of pointer pr2 to the
18      * pointer-to-pointer pr1
19      */
20     pr1 = &pr2;
21
22     /* Possible ways to find value of variable num*/
23     printf("\n Value of num is: %d", num);
24     printf("\n Value of num using pr2 is: %d", *pr2);
25     printf("\n Value of num using pr1 is: %d", **pr1);
26
```



C Pointers



Example of double Pointer

The output of the program shown above.

```
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ gcc p2p_1.c -o p2p
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ ./p2p

Value of num is: 123
Value of num using pr2 is: 123
Value of num using pr1 is: 123
Address of num is: 0x7ffe4f20ea54
Address of num using pr2 is: 0x7ffe4f20ea54
Address of num using pr1 is: 0x7ffe4f20ea54
Value of Pointer pr2 is: 0x7ffe4f20ea54
Value of Pointer pr2 using pr1 is: 0x7ffe4f20ea54
Address of Pointer pr2 is:0x7ffe4f20ea58
Address of Pointer pr2 using pr1 is:0x7ffe4f20ea58
Value of Pointer pr1 is:0x7ffe4f20ea58
Address of Pointer pr1 is:0x7ffe4f20ea60
(base) grivis@Grivis-Main:~/Documents/Active/TimS$
```



C Pointers



Example of double Pointer

```
#include <stdio.h>
int main()
{
    int num=123;

    //A normal pointer pr2
    int *pr2;

    //This pointer pr2 is a double pointer
    int **pr1;

    /* Assigning the address of variable num to the
     * pointer pr2
     */
    pr2 = &num;

    /* Assigning the address of pointer pr2 to the
     * pointer-to-pointer pr1
     */
    pr1 = &pr2;

    /* Possible ways to find value of variable num*/
    printf("\n Value of num is: %d", num);
    printf("\n Value of num using pr2 is: %d", *pr2);
    printf("\n Value of num using pr1 is: %d", **pr1);

    /*Possible ways to find address of num*/
    printf("\n Address of num is: %p", &num);
    printf("\n Address of num using pr2 is: %p", pr2);
    printf("\n Address of num using pr1 is: %p", *pr1);

    /*Find value of pointer*/
    printf("\n Value of Pointer pr2 is: %p", pr2);
    printf("\n Value of Pointer pr2 using pr1 is: %p", *pr1);

    /*Ways to find address of pointer*/
    printf("\n Address of Pointer pr2 is:%p",&pr2);
    printf("\n Address of Pointer pr2 using pr1 is:%p",pr1);

    /*Double pointer value and address*/
    printf("\n Value of Pointer pr1 is:%p",pr1);
    printf("\n Address of Pointer pr1 is:%p",&pr1);

    putchar('\n');

    return 0;
}
```




C Pointers



Example of double Pointer

You can understand the program logic with these simple equations:

```
num == *pr2 == **pr1  
&num == pr2 == *pr1  
&pr2 == pr1
```




C Pointers



Passing pointer to a function in C





C Pointers



In this part of the lesson, we will learn how to pass a pointer to a function as an argument. To understand this concept you must have a basic idea of Pointers and functions in C programming.

Just like any other argument, pointers can also be passed to a function as an argument. Lets take an example to understand how this is done.

Try this same program without pointer, you would find that the bonus amount will not reflect in the salary, this is because the change made by the function would be done to the local variables of the function. When we use pointers, the value is changed at the address of variable

```
p2p_1.c x p2function.c x
1  #include <stdio.h>
2  void salaryhike(int *var, int b)
3  {
4      *var = *var+b;
5  }
6  int main()
7  {
8      int salary=0, bonus=0;
9      printf("Enter the employee current salary:");
10     scanf("%d", &salary);
11     printf("Enter bonus:");
12     scanf("%d", &bonus);
13     salaryhike(&salary, bonus);
14     printf("Final salary: %d", salary);
15     return 0;
16 }
```

```
#include <stdio.h>
void salaryhike(int *var, int b)
{
    *var = *var+b;
}
int main()
{
    int salary=0, bonus=0;
    printf("Enter the employee current salary:");
    scanf("%d", &salary);
    printf("Enter bonus:");
    scanf("%d", &bonus);
    salaryhike(&salary, bonus);
    printf("Final salary: %d", salary);
    return 0;
}
```



C Pointers



In this part of the lesson, we will learn how to pass a pointer to a function as an argument. To understand this concept you must have a basic idea of Pointers and functions in C programming.

Just like any other argument, pointers can also be passed to a function as an argument. Lets take an example to understand how this is done.

Try this same program without pointer, you would find that the bonus amount will not reflect in the salary, this is because the change made by the function would be done to the local variables of the function. When we use pointers, the value is changed at the address of variable

```
File Edit View Search Terminal Help
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ gcc p2function.c -o p2function
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ ./p2function
Enter the employee current salary:10000
Enter bonus:2000
Final salary: 12000(base) grivis@Grivis-Main:~/Documents/Active/TimS$
```



C Pointers



Example 2: Swapping two numbers using Pointers

This is one of the most popular example that shows how to swap numbers using call by reference.

Try this same program without pointers and you would see that the numbers are not swapped. The reason is same that we have seen above in the first example.

```
p2p_1.c x p2function.c x p2.swap.c x
1 #include <stdio.h>
2 void swapnum(int *num1, int *num2)
3 {
4     int tempnum;
5
6     tempnum = *num1;
7     *num1 = *num2;
8     *num2 = tempnum;
9 }
10 int main( )
11 {
12     int v1 = 11, v2 = 77 ;
13     printf("Before swapping:");
14     printf("\nValue of v1 is: %d", v1);
15     printf("\nValue of v2 is: %d", v2);
16
17     /*calling swap function*/
18     swapnum( &v1, &v2 );
19
20     printf("\nAfter swapping:");
21     printf("\nValue of v1 is: %d", v1);
22     printf("\nValue of v2 is: %d", v2);
23 }
```

```
#include <stdio.h>
void swapnum(int *num1, int *num2)
{
    int tempnum;

    tempnum = *num1;
    *num1 = *num2;
    *num2 = tempnum;
}
int main( )
{
    int v1 = 11, v2 = 77 ;
    printf("Before swapping:");
    printf("\nValue of v1 is: %d", v1);
    printf("\nValue of v2 is: %d", v2);

    /*calling swap function*/
    swapnum( &v1, &v2 );

    printf("\nAfter swapping:");
    printf("\nValue of v1 is: %d", v1);
    printf("\nValue of v2 is: %d", v2);
}
```



C Pointers



Example 2: Swapping two numbers using Pointers

This is one of the most popular example that shows how to swap numbers using call by reference.

Try this same program without pointers and you would see that the numbers are not swapped. The reason is same that we have seen above in the first example.

```
File Edit View Search Terminal Help
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ gcc p2.swap.c -o p2swap
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ ./p2swap
Before swapping:
Value of v1 is: 11
Value of v2 is: 77
After swapping:
Value of v1 is: 77
Value of v2 is: 11(base) grivis@Grivis-Main:~/Documents/Active/TimS$
```




C Pointers



C – Function Pointer with examples





C Pointers



In C programming language, we can have a concept of Pointer to a function known as function pointer in C. In this part of the lesson, we will learn how to declare a function pointer and how to call a function using this pointer.





C Pointers



How to declare a function pointer?

```
function_return_type(*Pointer_name)(function argument list)
```

For example:

```
double (*p2f)(double, char)
```

Here double is a return type of function, p2f is name of the function pointer and (double, char) is an argument list of this function. Which means the first argument of this function is of double type and the second argument is char type.





C Pointers



Lets understand this with the help of an example: Here we have a function **sum** that calculates the sum of two numbers and returns the sum. We have created a pointer **f2p** that points to this function, we are invoking the function using this function pointer **f2p**.

```
p2p_1.c x p2function.c x p2.swap.c x func_pointer.c x
1 int sum (int num1, int num2)
2 {
3     return num1+num2;
4 }
5 int main()
6 {
7
8     /* The following two lines can also be written in a single
9     * statement like this: void (*fun_ptr)(int) = &fun;
10    */
11    int (*f2p) (int, int);
12    f2p = sum;
13    //Calling function using function pointer
14    int op1 = f2p(10, 13);
15
16    //Calling function in normal way using function name
17    int op2 = sum(10, 13);
18
19    printf("Output1: Call using function pointer: %d",op1);
20    printf("\nOutput2: Call using function name: %d", op2);
21
22    return 0;
23 }
```

```
int sum (int num1, int num2)
{
    return num1+num2;
}
int main()
{

    /* The following two lines can also be written
    in a single
    * statement like this: void (*fun_ptr)(int) =
    &fun;
    */
    int (*f2p) (int, int);
    f2p = sum;
    //Calling function using function pointer
    int op1 = f2p(10, 13);

    //Calling function in normal way using function
    name
    int op2 = sum(10, 13);

    printf("Output1: Call using function pointer:
    %d",op1);
    printf("\nOutput2: Call using function name:
    %d", op2);

    return 0;
}
```



C Pointers



Lets understand this with the help of an example: Here we have a function `sum` that calculates the sum of two numbers and returns the sum. We have created a pointer `f2p` that points to this function, we are invoking the function using this function pointer `f2p`.

```
File Edit View Search Terminal Help
(base) grivis@Grivis-Main:~/Documents/Active/TimS$ ./func_pointer
Output1: Call using function pointer: 23
Output2: Call using function name: 23
(base) grivis@Grivis-Main:~/Documents/Active/TimS$
```



C Pointers



The End of the Lesson

Au revoir! Adios!
HEY HEY! *Good bye!* Arrivederci!
Ciao! Salut! Tschüss!
adeus!

