



Function Pointer in C

Last Updated: 05-09-2018

In C, like **normal data pointers** (int *, char *, etc), we can have pointers to functions. Following is a simple example that shows declaration and function call using function pointer.

```
#include <stdio.h>
// A normal function with an int parameter
// and void return type
void fun(int a)
{
    printf("Value of a is %d\n", a);
}

int main()
{
    // fun_ptr is a pointer to function fun()
    void (*fun_ptr)(int) = &fun;

    /* The above line is equivalent of following two
       void (*fun_ptr)(int);
       fun_ptr = &fun;
    */

    // Invoking fun() using fun_ptr
    (*fun_ptr)(10);

    return 0;
}
```

Output:

Value of a is 10

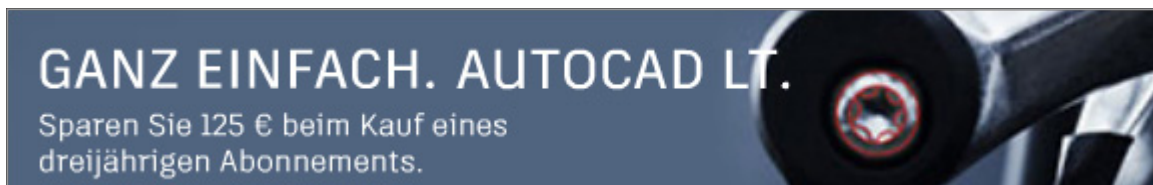
Why do we need an extra bracket around function pointers like fun_ptr in above example?

If we remove bracket, then the expression "void (*fun_ptr)(int)" becomes "void *fun_ptr(int)" which is declaration of a function that returns void pointer. See following post for details.



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



2) Unlike normal pointers, we do not allocate de-allocate memory using function pointers.

3) A function's name can also be used to get functions' address. For example, in the below program, we have removed address operator '&' in assignment. We have also changed function call by removing *, the program still works.

```
#include <stdio.h>
// A normal function with an int parameter
// and void return type
void fun(int a)
{
    printf("Value of a is %d\n", a);
}

int main()
{
    void (*fun_ptr)(int) = fun; // & removed

    fun_ptr(10); // * removed

    return 0;
}
```



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Value of a is 10



shows syntax for array of pointers.

5) Function pointer can be used in place of switch case. For example, in below program, user is asked for a choice between 0 and 2 to do different tasks.

```
#include <stdio.h>
void add(int a, int b)
{
    printf("Addition is %d\n", a+b);
}
void subtract(int a, int b)
{
    printf("Subtraction is %d\n", a-b);
}
void multiply(int a, int b)
{
    printf("Multiplication is %d\n", a*b);
}

int main()
{
    // fun_ptr_arr is an array of function pointers
    void (*fun_ptr_arr[])(int, int) = {add, subtract, multiply};
    unsigned int ch, a = 15, b = 10;

    printf("Enter Choice: 0 for add, 1 for subtract and 2 "
           "for multiply\n");
    scanf("%d", &ch);

    if (ch > 2) return 0;

    (*fun_ptr_arr[ch])(a, b);

    return 0;
}
```

Enter Choice: 0 for add, 1 for subtract and 2 for multiply

2

Multiplication is 150



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

returned from a function.



```
// A simple C program to show function pointers as parameter
#include <stdio.h>

// Two simple functions
void fun1() { printf("Fun1\n"); }
void fun2() { printf("Fun2\n"); }

// A function that receives a simple function
// as parameter and calls the function
void wrapper(void (*fun)())
{
    fun();
}

int main()
{
    wrapper(fun1);
    wrapper(fun2);
    return 0;
}
```

This point in particular is very useful in C. In C, we can use function pointers to avoid code redundancy. For example a simple `qsort()` function can be used to sort arrays in ascending order or descending or by any other order in case of array of structures. Not only this, with function pointers and void pointers, it is possible to use `qsort` for any data type.

```
// An example for qsort and comparator
#include <stdio.h>
#include <stdlib.h>

// A sample comparator function that is used
// for sorting an integer array in ascending order.
// To sort any array for any other data type and/or
// criteria, all we need to do is write more compare
// functions. And we can use the same qsort()
int compare (const void * a, const void * b)
{
    return ( *(int*)a - *(int*)b );
}

int main ()
```

```
qsort (arr, n, sizeof(int), compare);
```



```
}
```

Output:

```
5 10 12 15 80 90
```

Similar to qsort(), we can write our own functions that can be used for any data type and can do

Time to unleash full potential! Experience unlimited, unhindered learning with GeeksforGeeks Premium!

Only 500 slots available, [Grab Yours Now!](#)

for any data type. In fact we can use this search function to find close elements (below a threshold) by writing a customized compare function.

```
#include <stdio.h>
#include <stdbool.h>

// A compare function that is used for searching an integer
// array
bool compare (const void * a, const void * b)
{
    return ( *(int*)a == *(int*)b );
}

// General purpose search() function that can be used
// for searching an element *x in an array arr[] of
// arr_size. Note that void pointers are used so that
// the function can be called by passing a pointer of
// any type. ele_size is size of an array element
int search(void *arr, int arr_size, int ele_size, void *x,
          bool compare (const void * , const void *))
{
    // Since char takes one byte, we can use char pointer
    // for any type/ To get pointer arithmetic correct,
    // we need to multiply index with size of an array
    // element ele_size
    char *ptr = (char *)arr;

    int i;
    for (i=0; i<arr_size; i++)
        if (compare(ptr + i*ele_size, x))
            return i;

    // If element not found
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
int main()
{
```



```
        sizeof(int), &x, compare));
    return 0;
}
```

Output:

Returned index is 2

The above search function can be used for any data type by writing a separate customized compare().

7) Many object oriented features in C++ are implemented using function pointers in C. For example [virtual functions](#). Class methods are another example implemented using function pointers. Refer [this book](#) for more details.

Related Article: [Pointers in C and C++ | Set 1 \(Introduction, Arithmetic and Array\)](#)

References:

<http://www.cs.cmu.edu/~ab/15-123S11/AnnotatedNotes/Lecture14.pdf>

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/lecture-notes/MIT6_087IAP10_lec08.pdf

<http://www.cs.cmu.edu/~guna/15-123S11/Lectures/Lecture14.pdf>

This article is contributed by **Abhay Rathi**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Attention reader! Don't stop learning now. Get hold of all the important DSA concepts with the **DSA Self Paced Course** at a student-friendly price and become industry ready.

GigaCable Max
Macht den Unterschied:
1000 Mbit/s für
Dein Zuhause



Dauerhaft
günstig
ab 39,99 €
im Monat

Jetzt sichern



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



Recommended Posts:

[Double Pointer \(Pointer to Pointer\) in C](#)

[Pointer to an Array | Array Pointer](#)

[What is a Pointer to a Null pointer](#)

[How to declare a pointer to a function?](#)

[How to return a Pointer from a Function in C](#)

[Pointer vs Array in C](#)

['this' pointer in C++](#)

[Multidimensional Pointer Arithmetic in C/C++](#)

[When do we pass arguments by reference or pointer?](#)

[How to write C functions that modify head pointer of a Linked List?](#)

[Scope Resolution Operator Versus this pointer in C++?](#)

[Opaque Pointer](#)

[Passing by pointer Vs Passing by Reference in C++](#)

[void pointer in C / C++](#)

[NULL pointer in C](#)

[C | Pointer Basics | Question 1](#)

[C | Pointer Basics | Question 2](#)

[C | Pointer Basics | Question 3](#)

[C | Pointer Basics | Question 4](#)

[C | Pointer Basics | Question 17](#)

Article Tags : [C](#) [C-Pointers](#) [CPP-Functions](#) [cpp-pointer](#)

Practice Tags : [C](#)



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



3.5

☐ To-do ☐ DoneBased on **127** vote(s)[Improve Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)

 5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

 feedback@geeksforgeeks.org

Company

[About Us](#)[Careers](#)[Privacy Policy](#)[Contact Us](#)

Learn

[Algorithms](#)[Data Structures](#)[Languages](#)[CS Subjects](#)[Video Tutorials](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



@geeksforgeeks , Some rights reserved



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !