# Applications of Pointers in C/C++

Difficulty Level : Basic • Last Updated : 25 May, 2022

Prerequisite : Pointers in C/C++, Memory Layout of C Programs.

- **To pass arguments by reference**. Passing by reference serves two purposes

(i) **To modify variable of function in other.** Example to swap two variables;

**C**

```
// C program to demonstrate that we can change
// local values of one function in another using pointers.

#include <stdio.h>

void swap(int* x, int* y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

int main()
{
    int x = 10, y = 20;
    swap(&x, &y);
    printf("%d %d\n", x, y);
    return 0;
```

## C++

```
// C++ program to demonstrate that we can change
// local values of one function in another using
// pointers.
#include <iostream>
using namespace std;

void swap(int* x, int* y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

int main()
{
    int x = 10, y = 20;
    swap(&x, &y);
    cout << x << " " << y << endl;
    return 0;
}
```

Output :

```
20 10
```

(ii) **For efficiency purpose**. Example passing large structure without reference would create a copy of the structure (hence wastage of space).
Note : The above two can also be achieved through References in C++.

- **For accessing array elements.** Compiler internally uses pointers to access array elements.

## C

```
// C program to demonstrate that compiler
```

```
#include <stdio.h>

int main()
{
    int arr[] = { 100, 200, 300, 400 };

    // Compiler converts below to *(arr + 2).
    printf("%d ", arr[2]);

    // So below also works.
    printf("%d\n", *(arr + 2));

    return 0;
}
```

## C++

```
// C++ program to demonstrate that compiler
// internally uses pointer arithmetic to access
// array elements.
#include <iostream>
using namespace std;

int main()
{
    int arr[] = { 100, 200, 300, 400 };

    // Compiler converts below to *(arr + 2).
    cout << arr[2] << " ";

    // So below also works.
    cout << *(arr + 2) << " ";

    return 0;
}
```

Output :

```
300 300
```

- **To return multiple values.** Example returning square and square root of numbers.

## C

```c
// C program to demonstrate that using a pointer
// we can return multiple values.

#include <math.h>
#include <stdio.h>

void fun(int n, int* square, double* sq_root)
{
    *square = n * n;
    *sq_root = sqrt(n);
}

int main()
{

    int n = 100;
    int sq;
    double sq_root;
    fun(n, &sq, &sq_root);

    printf("%d %f\n", sq, sq_root);
    return 0;
}
```

## C++

```cpp
// C++ program to demonstrate that using a pointer
// we can return multiple values.
#include <bits/stdc++.h>
using namespace std;

void fun(int n, int* square, double* sq_root)
{
    *square = n * n;
    *sq_root = sqrt(n);
}

int main()
{

    int n = 100;
    int* sq = new int;
    cout << *sq << " " << *sq_root;
    return 0;
}
```

Output :

See a successful future with C++
A COMPLETE BEGINNER TO ADVANCED C++ COURSE
25+ Hours Of Content
150+ Curated Problems
24x7 Doubt Support
Explore Course

```
10000 10
```

- **Dynamic memory allocation** : We can use pointers to dynamically allocate memory. The advantage of dynamically allocated memory is, it is not deleted until we explicitly delete it.

**C**

```
// C program to dynamically allocate an
// array of given size.

#include <stdio.h>
#include <stdlib.h>
int* createArr(int n)
{
    int* arr = (int*)(malloc(n * sizeof(int)));
    return arr;
```

```
{
    int* pt = createArr(10);
    return 0;
}
```

**C++**

```
// C++ program to dynamically allocate an
// array of given size.
#include <iostream>
using namespace std;

int* createArr(int n)
{
    return new int[n];
}

int main()
{
    int* pt = createArr(10);
    return 0;
}
```

## Some Questions Regarding Pointers:

1. *What are the uses of a pointer?*
   Ans. Pointer is used in the following cases
   i) It is used to access array elements
   ii) It is used for dynamic memory allocation.
   iii) It is used in Call by reference
   iv) It is used in data structures like trees, graph, linked list etc.

2. *Are pointers integer?*
   Ans. No, pointers are not integers. A pointer is an address and a positive number.

3. *What does the error 'Null Pointer Assignment' means and what causes this error?*
   Ans. As null pointer points to nothing so accessing a uninitialized pointer or invalid location may cause an error.

Start Your Coding Journey Now!

   I. Static memory allocation
   II. Dynamic memory allocation

5. *What is pointer to a pointer?*
   Ans. If a pointer variable points another pointer value. Such a situation is known as a pointer to a
   pointer.
   Example:
     int *p1,**p2,v=10;
     P1=&v; p2=&p1;
     Here p2 is a pointer to a pointer

6. *What is an array of pointers?*
   Ans: If the elements of an array are addresses, such an array is called an array of pointers.

- **To implement data structures.**
  Example linked list, tree, etc. We cannot use C++ references to implement these data structures because references are fixed to a location (For example, we can not traverse a linked list using references)
- **To do system level programming where memory addresses are useful**. For example shared memory used by multiple threads. For more examples, see IPC through shared memory, Socket Programming in C/C++, etc

Like 52

## Start Your Coding Journey Now!

Previous Next

## Related Articles

1. Difference between constant pointer, pointers to constant, and constant pointers to constants
2. Why Does C Treat Array Parameters as Pointers?
3. What are Wild Pointers? How can we avoid?
4. Declare a C/C++ function returning pointer to array of integer pointers
5. Smart Pointers in C++ and How to Use Them
6. C Pointers
7. C++ Pointers
8. Pointers vs References in C++
9. What are near, far and huge pointers?
10. Computing Index Using Pointers Returned By STL Functions in C++

**Article Contributed By :**

kartik

## Vote for difficulty

Current difficulty : Basic

Easy Normal Medium Hard Expert

**Improved By :** RishabhPrabhu, SimarpreetSinghAneja, sarthak8858, vijendraniyer, susobhanakhuli

**Article Tags :** C-Pointers, cpp-pointer, C Language, C++

**Practice Tags :** CPP, c-pointers

Improve Article Report Issue

Courses

## News

Top News

Technology

Work & Career

Business

Finance

Lifestyle

Knowledge

## Languages

Python

Java

CPP

Golang

C#

SQL

Kotlin

## Web Development

Web Tutorials

Django Tutorial

HTML

JavaScript

Bootstrap

ReactJS

NodeJS

## Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

tart Your Coding Journey Now!