

Dangling, Void , Null and Wild Pointers

Difficulty Level : Easy • Last Updated : 28 Jun, 2021

Dangling pointer

A pointer pointing to a memory location that has been deleted (or freed) is called dangling pointer. There are **three** different ways where Pointer acts as dangling pointer

1. De-allocation of memory

```
// Deallocating a memory pointed by ptr causes
// dangling pointer
#include <stdlib.h>
#include <stdio.h>
int main()
{
    int *ptr = (int *)malloc(sizeof(int));

    // After below free call, ptr becomes a
    // dangling pointer
    free(ptr);

    // No more a dangling pointer
    ptr = NULL;
}
```

2. Function Call

```
// The pointer pointing to local variable becomes
// dangling when local variable is not static.
#include<stdio.h>
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Driver Code
int main()
{
    int *p = fun();
    fflush(stdin);

    // p points to something which is not
    // valid anymore
    printf("%d", *p);
    return 0;
}
```

Output:

A garbage Address

The above problem doesn't appear (or p doesn't become dangling) if x is a static variable.

```
// The pointer pointing to local variable doesn't
// become dangling when local variable is static.
#include<stdio.h>

int *fun()
{
    // x now has scope throughout the program
    static int x = 5;

    return &x;
}

int main()
{
    int *p = fun();
    fflush(stdin);

    // Not a dangling pointer as it points
    // to static variable.
    printf("%d", *p);
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
{
    int *ptr;
    .....
    .....
    {
        int ch;
        ptr = &ch;
    }
    .....
    // Here ptr is dangling pointer
}
```

[Void pointer](#)

Void pointer is a specific pointer type – void * – a pointer that points to some data location in storage, which doesn't have any specific type. Void refers to the type. Basically the type of data that it points to is can be any. If we assign address of char data type to void pointer it will become char Pointer, if int data type then int pointer and so on. Any pointer type is convertible to a void pointer hence it can point to any value.

Important Points

1. void pointers **cannot be dereferenced**. It can however be done using typecasting the void pointer
2. Pointer arithmetic is not possible on pointers of void due to lack of concrete value and thus size



[Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Topic-wise Practice](#) [C++](#) [Java](#) [Python](#)

```
#include<stdlib.h>
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Simpler tools lead to happier developers. And
happier developers lead to better results

GET \$100 FREE CREDIT

[HIDE AD](#) • [AD VIA BUYSPELLADS](#)

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// *((int*)ptr) dereferences the typecasted
// void pointer variable.
printf("Integer variable is = %d", *( (int*) ptr) );

// void pointer is now float
ptr = &y;
printf("\nFloat variable is= %f", *( (float*) ptr) );

return 0;
}
```

Output:

```
Integer variable is = 4
Float variable is= 5.500000
```

Refer [void pointer article](#) for details.

NULL Pointer

NULL Pointer is a pointer which is pointing to nothing. In case, if we don't have address to be assigned to a pointer, then we can simply use NULL.

```
#include <stdio.h>
int main()
{
    // Null Pointer
    int *ptr = NULL;

    printf("The value of ptr is %p", ptr);
    return 0;
}
```

Output :

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge

Start Your Coding Journey Now!

[Login](#)[Register](#)

Wild pointer

A pointer which has not been initialized to anything (not even NULL) is known as wild pointer. The pointer may be initialized to a non-NULL garbage value that may not be a valid address.

```
int main()
{
    int *p;  /* wild pointer */

    int x = 10;

    // p is not a wild pointer now
    p = &x;

    return 0;
}
```

This article is contributed by **Spoorthi Arun**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



The banner features the GeeksforGeeks logo on the left. In the center, it says "Participate In Bi-Wizard Coding Contests" with "Bi-Wizard" in a large, stylized font. To the right is an illustration of a person with dark hair holding a laptop. Further right, an orange ribbon banner says "Exclusively For School Students". At the bottom, it states "Held On 26th Of Every Month".

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Simpler tools lead to happier developers. And
happier developers lead to better results

GET \$100 FREE CREDIT

[HIDE AD](#) • [AD VIA BUYSPELLADS](#)

Start Your Coding Journey Now!

[Login](#)[Register](#)

An Uncommon representation of array elements

RECOMMENDED ARTICLES

Page : [1](#) [2](#) [3](#)

- 01

Difference between Dangling pointer and Void pointer
12, Jul 21
- 02

What are Wild Pointers? How can we avoid?
19, Feb 10
- 03

Difference between constant pointer, pointers to constant, and constant pointers to constants
26, Apr 21
- 04

Difference between "int main()" and "int main(void)" in C/C++?
07, Jan 14
- 05

How does 'void*' differ in C and C++?
08, Mar 10
- 06

Understanding "static" in "public static void main" in Java
10, Sep 18
- 07

Is it fine to write void main() or main() in C/C++?
05, Jan 14
- 08

Return From Void Functions in C++
03, Jul 16

Article Contributed By :



GeeksforGeeks

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Simpler tools lead to happier developers. And happier developers lead to better results

GET \$100 FREE CREDIT

[HIDE AD](#) • [AD VIA BUYSPELLADS](#)

Start Your Coding Journey Now!

[Login](#)[Register](#)

Improved By : [ashoklb](#), [sharathmaidargi](#)

Article Tags : [C-Pointers](#), [cpp-pointer](#), [C Language](#), [C++](#), [School Programming](#)

Practice Tags : [CPP](#)

[Improve Article](#)[Report Issue](#)

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](#)[In Media](#)[Contact Us](#)

Learn

[Algorithms](#)[Data Structures](#)[SDE Cheat Sheet](#)[Machine learning](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Simpler tools lead to happier developers. And
happier developers lead to better results

GET \$100 FREE CREDIT

[HIDE AD](#) • [AD VIA BUYSSELLADS](#)

Start Your Coding Journey Now!

Login

Register

Business
Finance
Lifestyle

Golang
C#
SQL

Web Development

Web Tutorials
Django Tutorial
HTML
CSS
JavaScript
Bootstrap

Contribute

Write an Article
Improve an Article
Pick Topics to Write
Write Interview Experience
Internships
Video Internship

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Simpler tools lead to happier developers. And happier developers lead to better results

GET \$100 FREE CREDIT

HIDE AD • AD VIA BUYSSELLADS