

[Courses](#)[Login](#)[Suggest an Article](#)

Commonly Asked C Programming Interview Questions | Set 1

What is the difference between declaration and definition of a variable/function

Ans: Declaration of a variable/function simply declares that the variable/function exists somewhere in the program but the memory is not allocated for them. But the declaration of a variable/function serves an important role. And that is the type of the variable/function. Therefore, when a variable is declared, the program knows the data type of that variable. In case of function declaration, the program knows what are the arguments to that functions, their data types, the order of arguments and the return type of the function. So that's all about declaration. Coming to the definition, when we define a variable/function, apart from the role of declaration, it also allocates memory for that variable/function. Therefore, we can think of definition as a super set of declaration. (or declaration as a subset of definition). From this explanation, it should be obvious that a variable/function can be declared any number of times but it can be defined only once. (Remember the basic principle that you can't have two locations of the same variable/function).

```
// This is only declaration. y is not allocated memory by this statement
extern int y;
```

```
// This is both declaration and definition, memory to x is allocated by this statement
int x;
```

What are different storage class specifiers in C?

Ans: auto, register, static, extern

What is scope of a variable? How are variables scoped in C?

Ans: Scope of a variable is the part of the program where the variable may directly be accessible. In C, all identifiers are lexically (or statically) scoped. See [this](#) for more details.

How will you print "Hello World" without semicolon?

Ans:

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



```
int main(void)
{
    if (printf("Hello World")) {
    }
}
```

See print “Geeks for Geeks” without using a semicolon for answer.



When should we use pointers in a C program?

1. To get address of a variable
2. *For achieving pass by reference in C:* Pointers allow different functions to share and modify their local variables.
3. *To pass large structures* so that complete copy of the structure can be avoided.
4. *To implement “linked” data structures* like linked lists and binary trees.

What is NULL pointer?

Ans: NULL is used to indicate that the pointer doesn't point to a valid location. Ideally, we should initialize pointers as NULL if we don't know their value at the time of declaration. Also, we should make a pointer NULL when memory pointed by it is deallocated in the middle of a program.

What is Dangling pointer?

Ans: Dangling Pointer is a pointer that doesn't point to a valid memory location. Dangling pointers arise when an object is deleted or deallocated, without modifying the value of the pointer, so that the pointer still points to the memory location of the deallocated memory. Following are examples.

```
// EXAMPLE 1
int* ptr = (int*)malloc(sizeof(int));
.....free(ptr);

// ptr is a dangling pointer now and operations like following are invalid
*ptr = 10; // or printf("%d", *ptr);
```

```
// EXAMPLE 2
int* ptr = NULL
{
    int x = 10;
    ptr = &x;
}
// x goes out of scope and memory allocated to x is free now.
// So ptr is a dangling pointer now.
```



What is memory leak? Why it should be avoided

Ans: Memory leak occurs when programmers create a memory in heap and forget to delete it. Memory leaks are particularly serious issues for programs like daemons and servers which by definition never terminate.

```
/* Function with memory leak */
#include <stdlib.h>

void f()
{
    int* ptr = (int*)malloc(sizeof(int));

    /* Do some work */

    return; /* Return without freeing ptr*/
}
```

What are local static variables? What is their use?

Ans: A local static variable is a variable whose lifetime doesn't end with a function call where it is declared. It extends for the lifetime of complete program. All calls to the function share the same copy of local static variables. Static variables can be used to count the number of times a function is called. Also, static variables get the default value as 0. For example, the following program prints "0 1"

```
#include <stdio.h>
void fun()
{
    // static variables get the default value as 0.
    static int x;
    printf("%d ", x);
    x = x + 1;
}

int main()
{
    fun();
    fun();
    return 0;
}
// Output: 0 1
```

What are static functions? What is their use?

Ans: In C, functions are global by default. The "static" keyword before a function name makes it static. Unlike global functions in C, access to static functions is restricted to the file where they are declared. Therefore, when we want to restrict access to functions, we make them static. Another reason for making functions static can be reuse of the same function name in other files. See [this](#) for examples and more details.

- [Commonly Asked C Programming Interview Questions | Set 2](#)

- [Practices Quizzes on C](#)
- [C articles](#)

You may also like:

[Commonly Asked C Programming Interview Questions | Set 2](#)

[Commonly Asked Java Programming Interview Questions | Set 1](#)

[Amazon's most asked interview questions](#)

[Microsoft's most asked interview questions](#)

[Accenture's most asked Interview Questions](#)

[Commonly Asked OOP Interview Questions](#)

[Commonly Asked C++ Interview Questions](#)

[Commonly asked DBMS interview questions | Set 1](#)

[Commonly asked DBMS interview questions | Set 2](#)

[Commonly Asked Operating Systems Interview Questions | Set 1](#)

[Commonly Asked Data Structure Interview Questions.](#)

[Commonly Asked Algorithm Interview Questions](#)

[Commonly asked Computer Networks Interview Questions](#)

[Top 10 algorithms in Interview Questions](#)

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

Recommended Posts:

[Commonly Asked C Programming Interview Questions | Set 3](#)

[Commonly Asked C Programming Interview Questions | Set 2](#)

[Commonly Asked OOP Interview Questions | Set 1](#)

[Commonly Asked C++ Interview Questions | Set 1](#)

[Commonly Asked C++ Interview Questions | Set 2](#)

[Commonly asked JavaScript Interview Questions | Set 1](#)

[Commonly Asked Algorithm Interview Questions | Set 1](#)

[Commonly Asked Data Structure Interview Questions | Set 1](#)

[Commonly Asked Questions in Goldman Sachs Interviews](#)

[Amazon's most asked interview questions](#)

[Amazon's most frequently asked interview questions | Set 2](#)

[Frequently Asked Questions regarding Placements](#)

[MAQ Software most Frequently Asked Questions](#)

[10 Most asked Questions from Java Programmers](#)

[Top 20 Dynamic Programming Interview Questions](#)



Improved By : [moosavi001](#)

Article Tags : [C](#) [interview-preparation](#) [placement preparation](#)

Practice Tags : [C](#)



9

☐ To-do ☐ Done

2.3

Based on **23** vote(s)

[Feedback/ Suggest Improvement](#)

[Add Notes](#)

[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](#)

[Share this post!](#)



A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved

