Yamu Poudel's

# Pointers & Memory Allocation

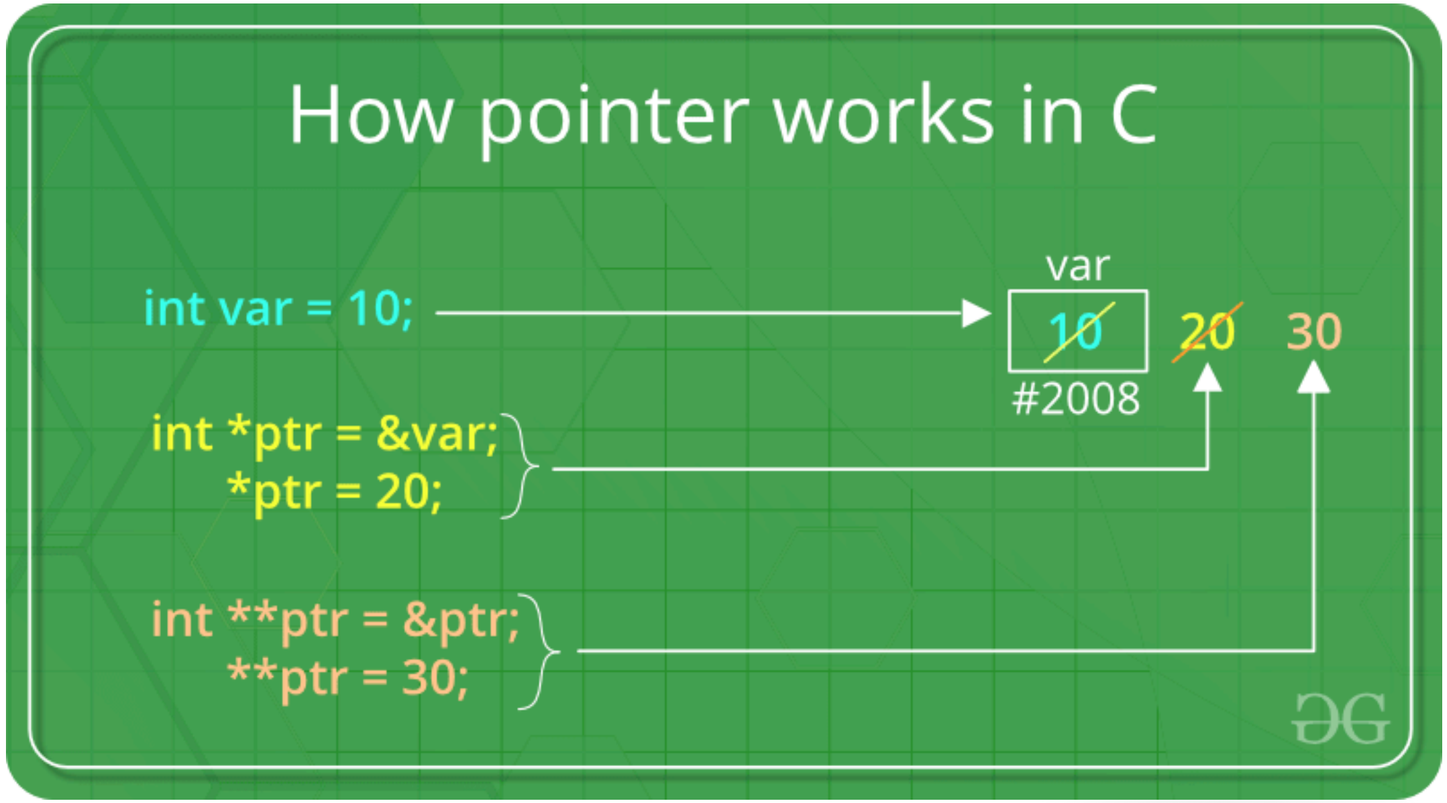
High Performance Computing


# Why Pointers?

Pointers reduce the length and complexity of a program. Pointers make possible to return more than one value from the function. Pointers increase the processing speed. In other words, Execution time with pointers is faster because data are manipulated with the address, that is, direct access to memory location.

Yamu's Class

# This is How Pointers Works!



# Real-Time Usage Of Pointers

Points for discussion

To pass arguments by reference.

For accessing array elements.

Dynamic memory allocation.

To do system-level programming where memory addresses are useful

# Drawbacks Of Pointers

if pointers are pointed to some incorrect location then it may end up reading a wrong value.

Pointers are slower than normal variable

If we forgot to deallocate a memory then it will lead to a memory leak.

It requires one additional dereferences step

If sufficient memory is not available during runtime for the storage of pointers, the program may crash


I KNOW A GUY WHO KNOWS A GUY
WHO KNOWS A GUY

# Types Of Pointers

Pointers are of three types mainly called a null pointer, Void pointer, and wild pointer. Let's find out more about each of them in detail.

### NULL POINTERS

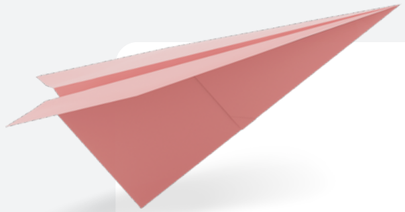
A null pointer is nothing but a command used to direct to an empty location in a computer system. It doesn't point to any variable or function and is also used to denote the ending of a memory search.

### VOID POINTERS

It is a form of pointer which is used to point another variable of any data type. So, it can store the address of a variable of any data type.

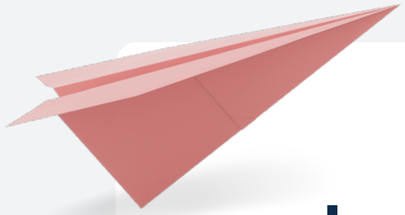
### WILD POINTER

A Pointer in C that has not been initialized till its first use is known as Wild pointer. A wild pointer points to some random memory location.



# Memory Allocation

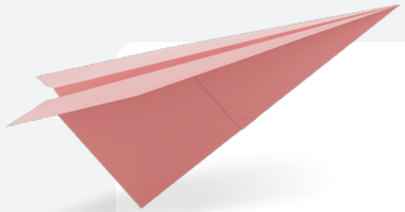
In C, dynamic memory is allocated from the heap using some standard library functions.



In certain programming languages including C and Pascal, a heap is an area of pre-reserved computer main storage ( memory ) that a program process can use to store data in some variable amount that won't be known until the program is running.

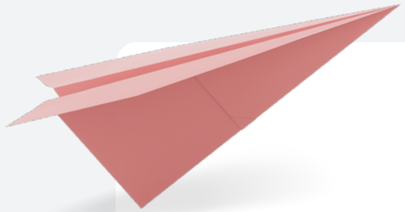
Use dynamic in the following situations:

1. When you need a lot of memory. The typical stack size is 1 MB, so anything bigger than 50-100KB should better be dynamically allocated, or you're risking a crash. Some platforms can have this limit even lower.
2. When the memory must live after the function returns. Stack memory gets destroyed when the function ends, dynamic memory is freed when you want.
3. When you're building a structure (like an array, or graph) of the size that is unknown (i.e. may get big), dynamically changes, or is too hard to precalculate. Dynamic allocation allows your code to naturally request memory piece by piece at any moment and only when you need it. It is not possible to repeatedly request more and more stack space in a for loop.



# malloc()

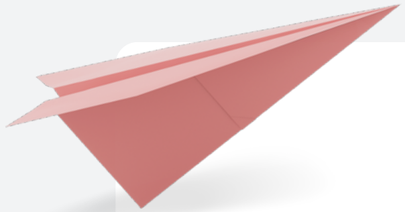
The “malloc” or “memory allocation” method in C is used to dynamically allocate a single large block of memory with the specified size. It returns a pointer of type void which can be cast into a pointer of any form.



# calloc()

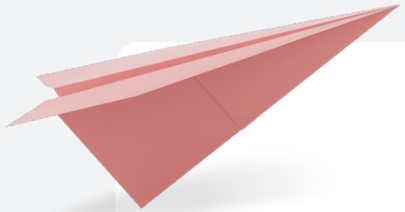
“calloc” or “contiguous allocation” method in C is used to dynamically allocate the specified number of blocks of memory of the specified type. it is very much similar to malloc() but has two different points and these are:

1. It initializes each block with a default value ‘0’.
2. It has two parameters or arguments as compare to malloc().



# free()

“free” method in C is used to dynamically de-allocate the memory. The memory allocated using functions malloc() and calloc() is not de-allocated on their own. Hence the free() method is used, whenever the dynamic memory allocation takes place. It helps to reduce wastage of memory by freeing it.



# realloc()

“realloc” or “re-allocation” method in C is used to dynamically change the memory allocation of a previously allocated memory. In other words, if the memory previously allocated with the help of malloc or calloc is insufficient, realloc can be used to dynamically re-allocate memory.

## malloc()

- It is a function that creates one block of memory of a fixed size.
- It only takes one argument
- It is faster than calloc.
- It has high time efficiency.
- It is used to indicate memory allocation

## calloc()

- It is a function that assigns more than one block of memory to a single variable.
- It takes two arguments.
- It is slower than malloc().
- It has low time efficiency.
- It is used to indicate contiguous memory allocation


BLAH

# Do you have any questions?

I hope you learned something new.


GOOD
Have a Great Day
day