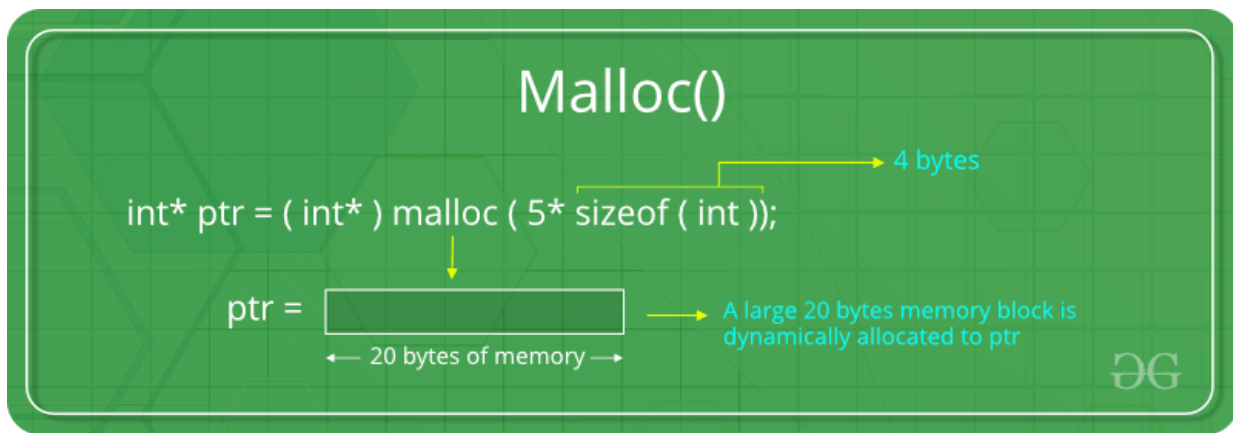


Dynamic Memory Allocation

- **Dynamic Memory Allocation** can be defined as a procedure of allocating memory at the runtime.
- There are 4 library functions provided by C defined under `<stdlib.h>` header file to facilitate dynamic memory allocation in C programming. They are:
 - `malloc()`
 - `calloc()`
 - `free()`
 - `realloc()`
- **malloc:**
 - “malloc” or “memory allocation” method 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.
 - It initializes each block with default garbage value.
 - If space is insufficient, allocation fails and returns a NULL pointer.
 - **Syntax:**
 - `ptr = (cast-type*) malloc (byte-size)`
 - **Example:**
 - `ptr = (int*) malloc (100 * sizeof(int));`
Explanation: *Since the size of int is 4 bytes, this statement will allocate 400 bytes of memory. And the pointer ptr holds the address of the first byte in the allocated memory.*



- **calloc:**

- “calloc” or “contiguous allocation” method is used to dynamically allocate the specified number of blocks of memory of the specified type.
- It initializes each block with a default value ‘0’.
- If space is insufficient, allocation fails and returns a NULL pointer.

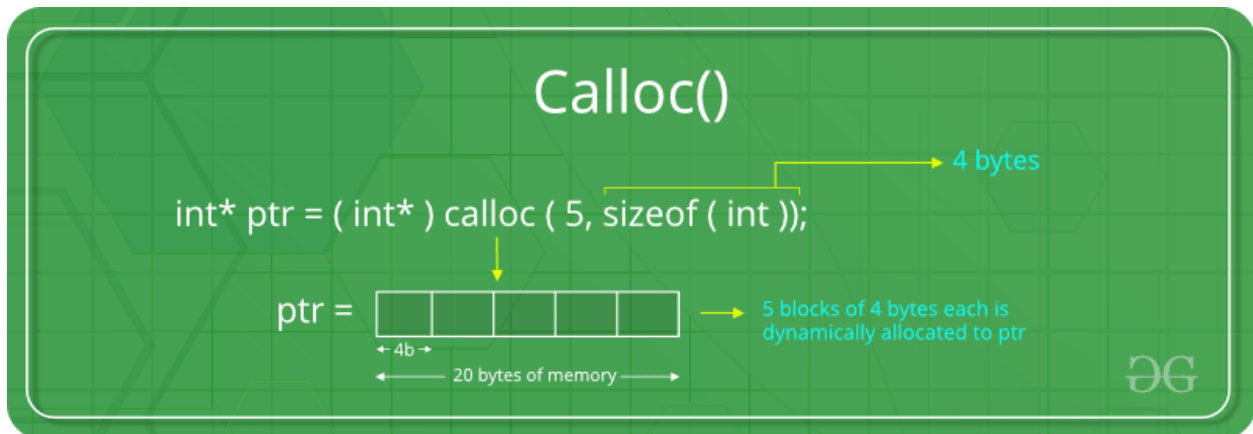
- **Syntax:**

- `ptr = (cast-type*) calloc (n, element-size);`

- **Example:**

- `ptr = (float*) calloc (25, sizeof(float));`

Explanation: *This statement allocates contiguous space in memory for 25 elements each with the size of the float.*



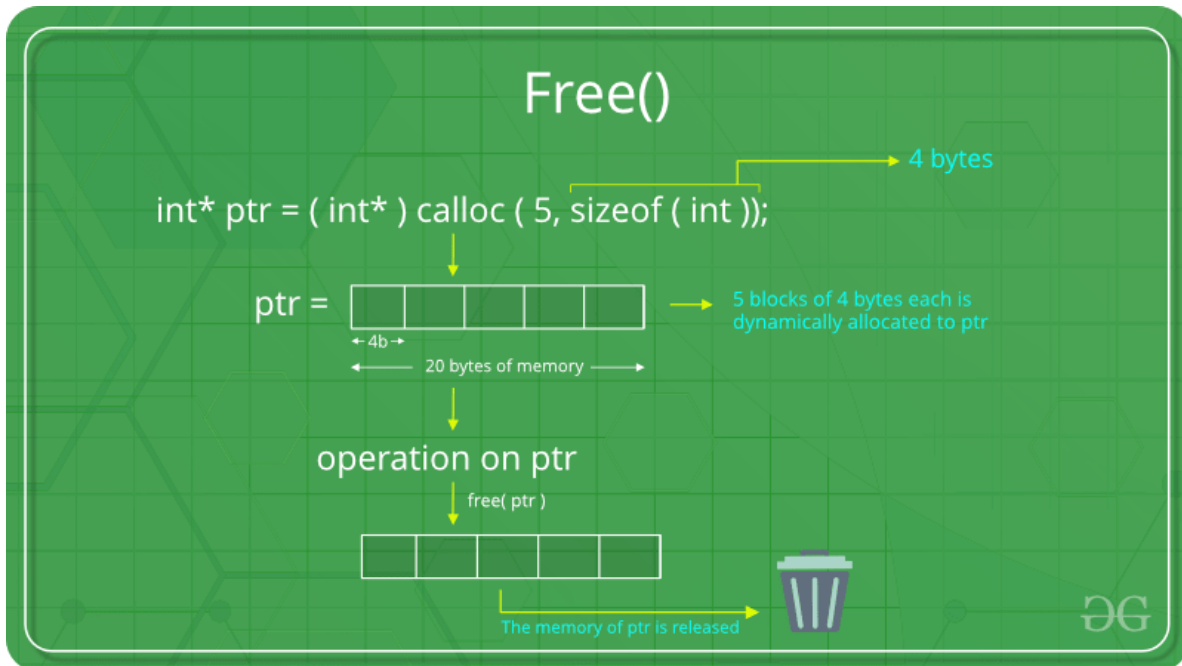
- **free:**

- “free” method 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.

- **Syntax:**

- `free(ptr);`

- How does free() know the size of memory to be deallocated?
 - When we use the dynamic memory allocation techniques for memory allocations, then this is done in the actual heap section. It creates one word larger than the requested size.
 - This extra word is used to store the size. This size is used by free() when it wants to clear the memory space.



- **realloc:**

- “realloc” or “re-allocation” method is used to dynamically change the memory allocation of a previously allocated memory by malloc or calloc.
- 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.
- Re-allocation of memory maintains the already present value and new blocks will be initialized with default garbage value.
- If space is insufficient, allocation fails and returns a NULL pointer.
- **Syntax:**
 - `realloc(ptr, newSize);`
Explanation: Where *ptr* is reallocated with new size '*newSize*'.

Realloc()

