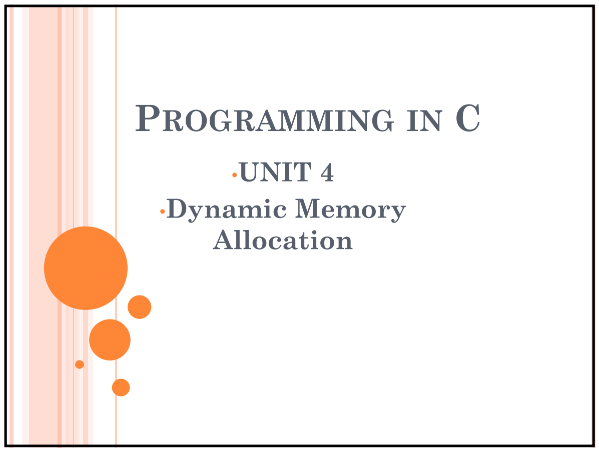
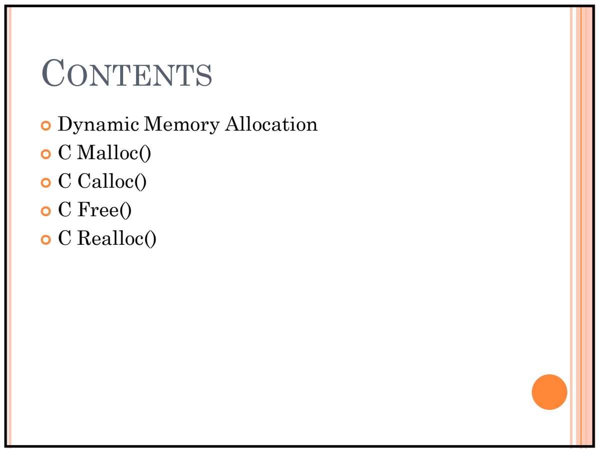


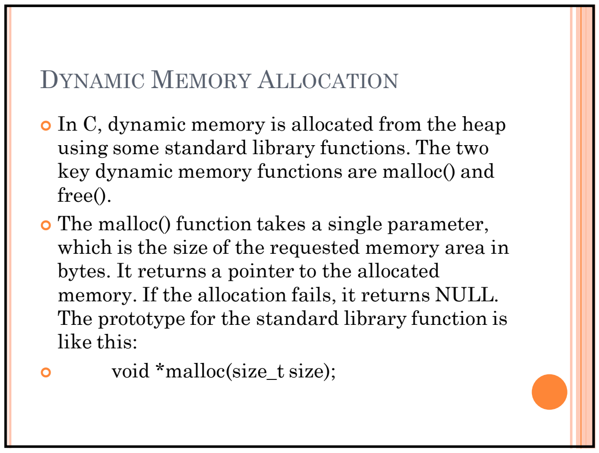
Programming in C

- UNIT 4
- Dynamic Memory Allocation



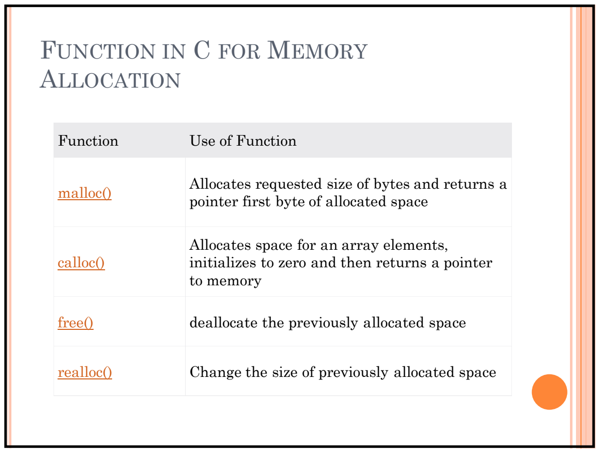
Contents

- Dynamic Memory Allocation
- C Malloc()
- C Calloc()
- C Free()
- C Realloc()



Dynamic Memory Allocation

- In C, dynamic memory is allocated from the heap using some standard library functions. The two key dynamic memory functions are malloc() and free().
- The malloc() function takes a single parameter, which is the size of the requested memory area in bytes. It returns a pointer to the allocated memory. If the allocation fails, it returns NULL. The prototype for the standard library function is like this:
- void *malloc(size_t size);



Function in C for Memory Allocation

Function	Use of Function
malloc()	Allocates requested size of bytes and returns a pointer first byte of allocated space
calloc()	Allocates space for an array elements, initializes to zero and then returns a pointer to memory
free()	deallocate the previously allocated space
realloc()	Change the size of previously allocated space

C MALLOC()

- The name malloc stands for "memory allocation".
- The function malloc() reserves a block of memory of specified size and return a pointer of type void which can be casted into pointer of any form.
- **Syntax of malloc()**
- ptr = (cast-type*) malloc(byte-size)Here, ptr is pointer of cast-type. The malloc() function returns a pointer to an area of memory with size of byte size. If the space is insufficient, allocation fails and returns NULL pointer.
- ptr = (int*) malloc(100 * sizeof(int));
- This statement will allocate either 200 or 400 according to size of int 2 or 4 bytes respectively and the pointer points to the address of first byte of memory.

C CALLOC()

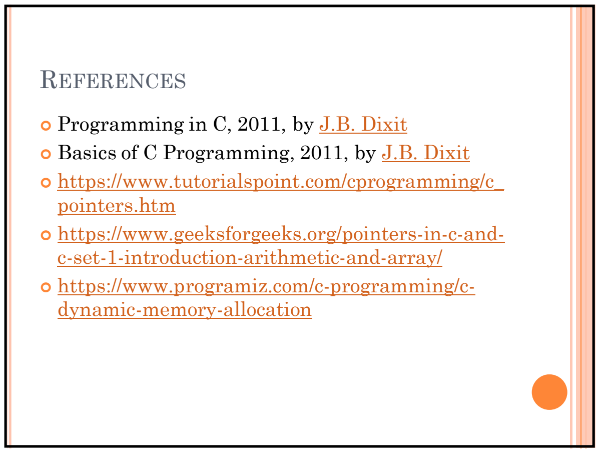
- The name calloc stands for "contiguous allocation".
- The only difference between malloc() and calloc() is that, malloc() allocates single block of memory whereas calloc() allocates multiple blocks of memory each of same size and sets all bytes to zero.
- **Syntax of calloc()**
- ptr = (cast-type*)calloc(n, element-size);This statement will allocate contiguous space in memory for an array of n elements. For example:
- ptr = (float*) calloc(25, sizeof(float));This statement allocates contiguous space in memory for an array of 25 elements each of size of float, i.e, 4 bytes.

C FREE()

- Dynamically allocated memory created with either calloc() or malloc() doesn't get freed on its own. You must explicitly use free() to release the space.
- **syntax of free()**
- free(ptr);This statement frees the space allocated in the memory pointed by ptr.

C REALLOC()

- If the previously allocated memory is insufficient or more than required, you can change the previously allocated memory size using realloc().
- **Syntax of realloc()**
- ptr = realloc(ptr, newsize);Here, ptr is reallocated with size of newsize.



References

- Programming in C, 2011, by J.B. Dixit
- Basics of C Programming, 2011, by J.B. Dixit
- https://www.tutorialspoint.com/cprogramming/c_pointers.htm
- https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/
- https://www.programiz.com/c-programming/c-dynamic-memory-allocation