Presented By: Mostafa Saqly

C Dynamic Memory Allocation

Content

- C Dynamic Memory Allocation
- pointerVariable=(casttype*)malloc(size*sizeof(datatype));

C Dynamic Memory Allocation

- r C program using standard library functions: malloc(), calloc(), free() and realloc().
- As you know, an array is a collection of a fixed number of values. Once the size of an array is declared, you cannot change it.
- Sometimes the size of the array you declared may be insufficient. To solve this issue, you can allocate memory manually during run-time. This is known as dynamic memory allocation in C programming.
- To allocate memory dynamically, library functions are malloc(), calloc(), realloc() and free() are used. These functions are defined in the <stdlib.h> header file.

C Dynamic Memory Allocation

C malloc()

- The name "malloc" stands for memory allocation.
- The malloc() function reserves a block of memory of the specified number of bytes. And, it returns a pointer of void which can be casted into pointers of any form.
- **C calloc()**
- The name "calloc" stands for contiguous allocation.
- The malloc() function allocates memory and leaves the memory uninitialized. Whereas, the calloc() function allocates memory and initializes all bits to zero.

C Dynamic Memory Allocation

C free()

- Dynamically allocated memory created with either calloc() or malloc() doesn't get freed on their own. You must explicitly use free() to release the space.

C realloc()

- If the dynamically allocated memory is insufficient or more than required, you can change the size of previously allocated memory using the realloc() function.

Thanks!

Any questions?