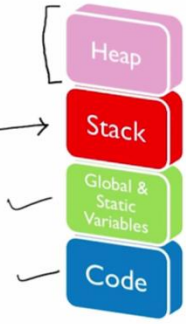
Dynamic Memory Allocation Malloc Calloc Realloc & Free()- C Tutorial In Hindi #47.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

DYNAMIC MEMORY ALLOCATION: RECAP

- An statically allocated variable or array has a fixed size in memory.
- **Dynamic Memory Allocation** is a way in which the size of a data structure can be changed during the runtime.
- Memory assigned to a program in a typical architecture can be broken down into four segments:
  1. Code
  2. Static/global variables
  3. Stack
  4. Heap

int arr[10];



01:41 29:52

Dynamic Memory Allocation Malloc Calloc Realloc & Free()- C Tutorial In Hindi #47.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

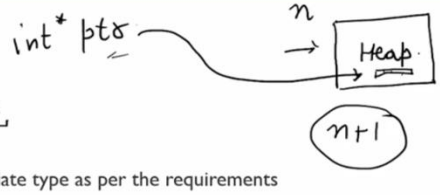
FUNCTIONS FOR DYNAMIC MEMORY ALLOCATION IN C

- In Dynamic memory allocation, the memory is allocated at runtime from the heap segment
- We have four functions that help us achieve this task:
  - malloc
  - calloc
  - realloc
  - free

→ Dynamic memory allocation

02:59 29:52

C MALLOC()

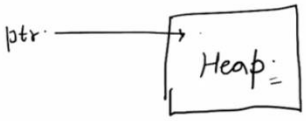


- malloc() stands for memory allocation
- It reserves a block of memory with the given amount of bytes.
- The return value is a void pointer to the allocated space
- Therefore the void pointer needs to be casted to the appropriate type as per the requirements
- However, if the space is insufficient, allocation of memory fails and it returns a NULL pointer.
- All the values at allocated memory are initialized to garbage values
  - Syntax:

```
ptr = (ptr-type*) malloc(size_in_bytes);
```

```
int * ptr;
ptr = (int*) malloc (3 * sizeof(int))
```


C CALLOC()

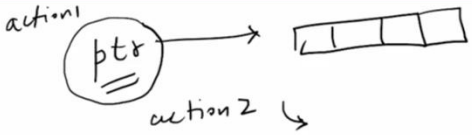


- calloc() stands for contiguous allocation
- It reserves n blocks of memory with the given amount of bytes.
- The return value is a void pointer to the allocated space
- Therefore the void pointer needs to be casted to the appropriate type as per the requirements
- However, if the space is insufficient, allocation of memory fails and it returns a NULL pointer.
- All the values at allocated memory are initialized to 0
  - Syntax:

```
ptr = (ptr-type*) calloc(n, size_in_bytes);
```

```
1 or    int
```

C REALLOC()



- realloc() stands for reallocation
- If the dynamically allocated memory is insufficient we can change the size of previously allocated memory using realloc() function
- Syntax:

```
ptr = (ptr-type*) realloc(ptr, new_size_in_bytes);
```

36 bytes.

40 bytes

C FREE()

ptr = (int*) malloc(3 * sizeof(

- free() is used to free the allocated memory
- If the dynamically allocated memory is not required anymore, we can free it using free function.
- This will free the memory being used by the program in the heap
- Syntax:

```
free(ptr);
```