

Assignment - 18

Q1] What is the difference between static memory allocation & dynamic memory allocation?

Ans

1] Static :- No need of specific function for memory allocation.
Dynamic :- We have to use specific functions for memory allocation & deallocation.

2] Static :- The calculation of memory allocation is done at compile-time.

Dynamic :- The calculation of "memory allocation" is done at run-time.

3] Static :- It may lead to problem of memory shortage or memory wastage.

Dynamic :- There is no such problem of memory shortage or memory wastage.

4] Static :- We can't deallocate memory after its use gets completed.

Dynamic :- We can deallocate the memory after its use gets completed.

5] Static :- There is no memory allocation feature at run-time.

Dynamic :- Dynamic memory allocation may fail ~~due to~~ at run-time due to shortage of memory.

7] Static :- It is a fast way of memory allocation as there is no calculation at run-time.

Dynamic :- It is slow as it needs calculations at run time.

8] Static :- the static memory allocation gets allocated inside the memory locations based on storage classes (either static or data segment or static segment or CPU register)

Dynamic :- In case of dynamic memory allocation the ~~the~~ memory gets allocated inside heap section (In C++ heap is also called free store)

Q2] What are the advantages & disadvantages of dynamic memory allocation over static memory allocation?

Ans

1] Advantages Of Dynamic memory allocation

1] There is no problem of memory shortage or memory wastage

1] We can deallocate memory after its use gets completed.

2] Disadvantages :

- 1] It is slow as it ~~nee~~ does calculation for memory allocation at runtime
- 2] Dynamic memory allocation may fail at runtime due to shortage of memory.

Q3] Which functions are used in C programming for dynamic memory allocations & deallocations?

Ans:

- | | | |
|------------|---|--------------|
| 1] malloc | } | Allocation |
| 2] calloc | | |
| 3] realloc | | |
| 4] free | } | Deallocation |
| | | |

Q4] Which functions/operators are used in C++ for dynamic memory allocations & deallocations?

Ans:

1] Conceptually all ~~prev~~ points of dynamic memory allocation are exactly same in case of C++ & Java.

2] In case of C++ we use different syntax.

3] In C++ to allocate memory dynamically we use the operator 'new' & to deallocate that memory we use 'delete' operator.

Example

```
int main()
```

```
{
```

```
    int arr[5]; // static
```

int *ptr1 = (int *) malloc (5 * sizeof (int)) // In C

int * ptr2 = new int [5]; // In C++

free (ptr1); // In C

delete [] ptr2;

Q 4] The new operator internally calls the malloc function

5] There is no need of typecasting in case of new.

6] There is no resize activity allowed in case of C++. (There is no virtual function like realloc)

Q 5]

What is difference between malloc & calloc?

Ans malloc & calloc both are used for dynamic memory allocations.

1] In case of malloc we have to pass one parameter, i.e. number of bytes we want to allocate.

2] In case of calloc we have to pass two parameters

3] First parameter of calloc is No. of elements & second parameter of calloc is size of each element

Prototype of malloc

void * malloc (int size);

Prototype of calloc

void * calloc (.

PA/E No.	
DATE	/ /

Prototype of calloc

`void * calloc (int no-of-elements, int size-of-each-element);`

Q] malloc & calloc work same internally.

Ans] Generally calloc is used for allocating memory for array.

Q6] Explain the prototype of malloc function?

Ans

1] malloc is a function used for allocation of memory dynamically

2] Prototype of malloc is :

`void * malloc (int size);`

1] The return value of malloc is void* which indicates the address of allocated memory

2] After getting the address we have to typecast, depending on our requirement.

3] The malloc function accepts only one parameter i.e the number of bytes we want to allocate.

Q7] Why return value of malloc, calloc & realloc function is void*?

Ans.] The return value of these functions is void because, as they are allocating memory without any type.

∴ As

Q8] What are the different uses of realloc function?

Ans

1] The realloc function is used to resize the allocated memory's size

2] ~~Re~~ realloc is used to increase or decrease the size of already allocated memory

Proto type of realloc

`void* realloc (void* ptr, int new-size)`

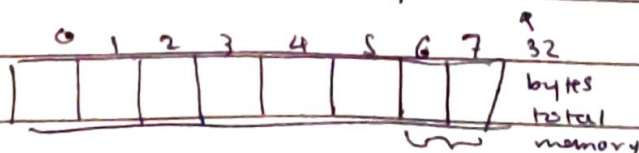
3] realloc can also be used to allocate or deallocate memory.



Memory allocated using malloc/calloc
`int* ptr = (int*) malloc (5 * size of (int))`

Increasing memory using realloc

`ptr = (int*) realloc (ptr, 32)`



Decreasing memory using realloc

`ptr = (int*) realloc (ptr, 16)`



Earlier memory = 24 bytes + 8 bytes = 32 bytes

Q7] What happens if the first parameter of `realloc` function is `NULL`?

Ans `int* ptr = (int*) realloc(NULL, 24)`

If the first parameter of `realloc` is `NULL` then the `realloc` works like `malloc`.

It returns the address zero & creates/allocates 24 bytes of memory in heap.

Q8] What happens if the second parameter of `realloc` function is 0?

Ans If the second parameter of `realloc` is 0 it acts as free function.

`int* ptr = (int*) realloc(ptr, 0)`
↑ ↑
location 0 bytes

Allocates 0 bytes where ptr points to.