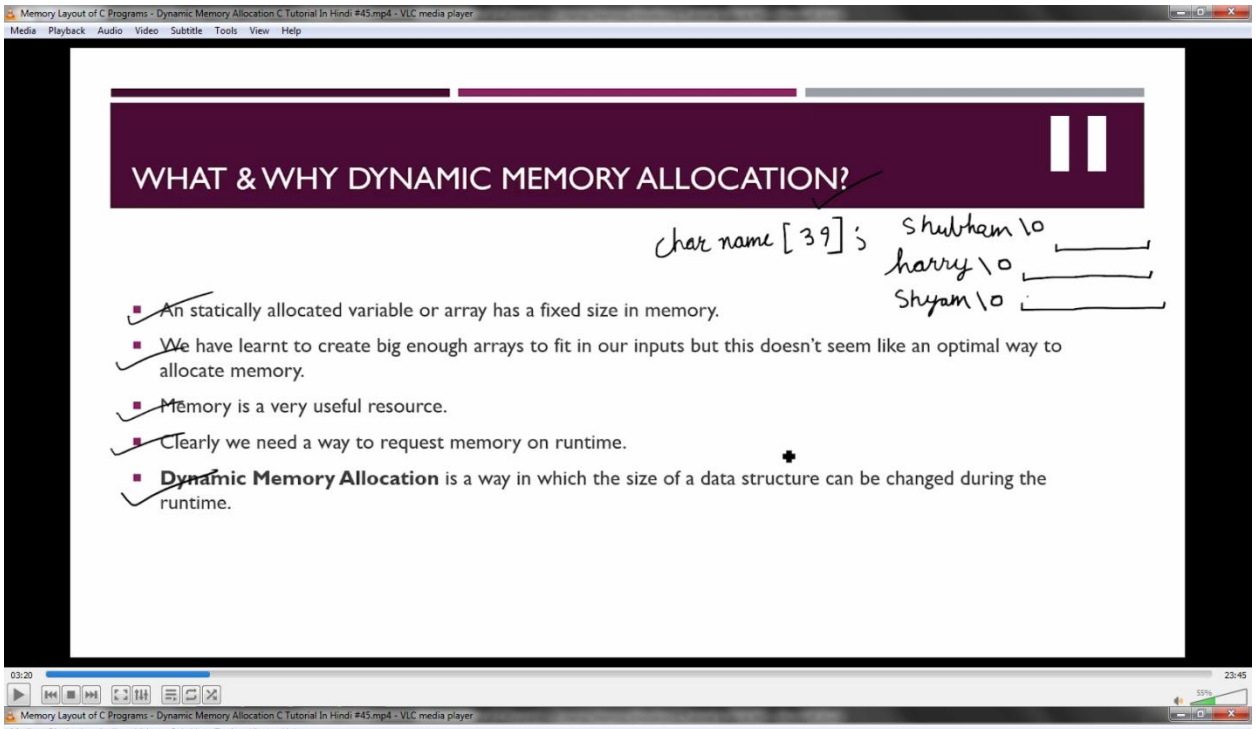
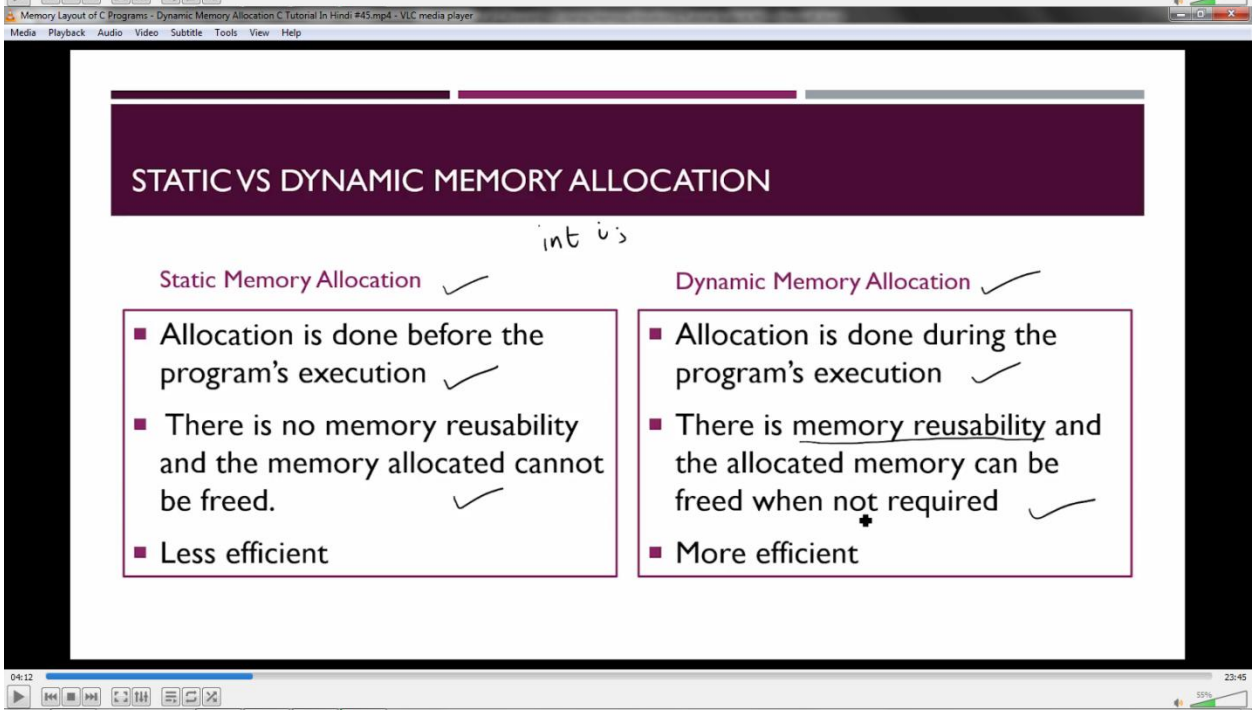
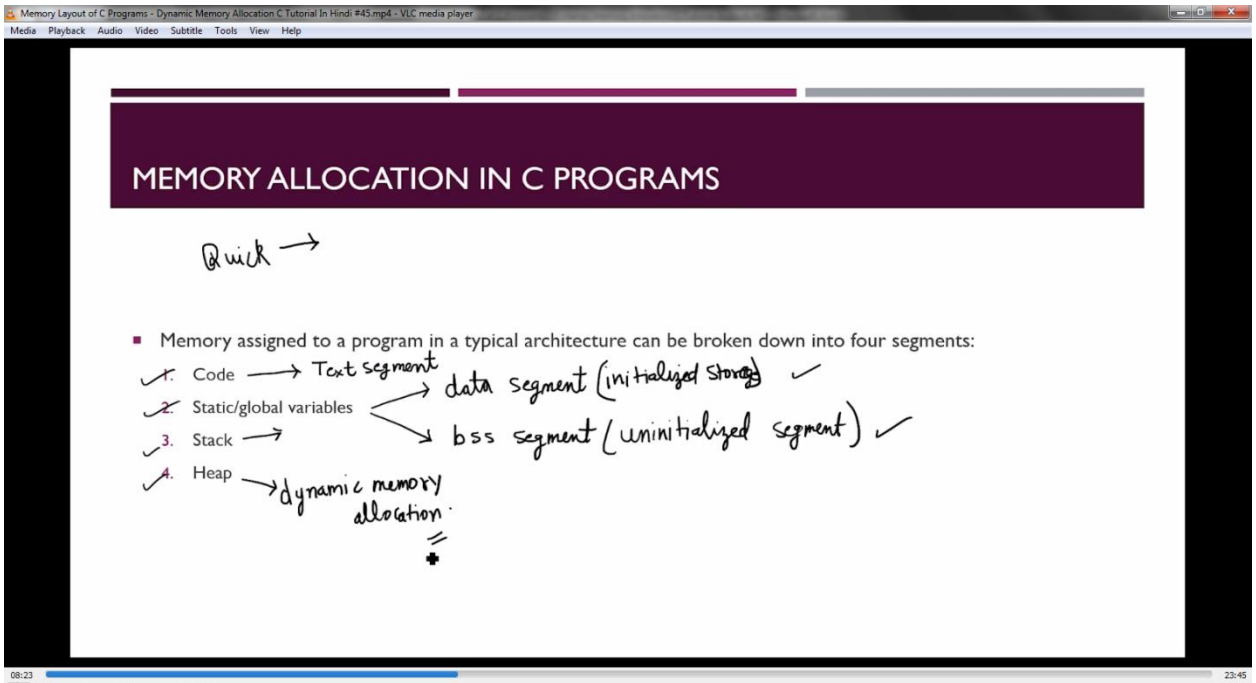
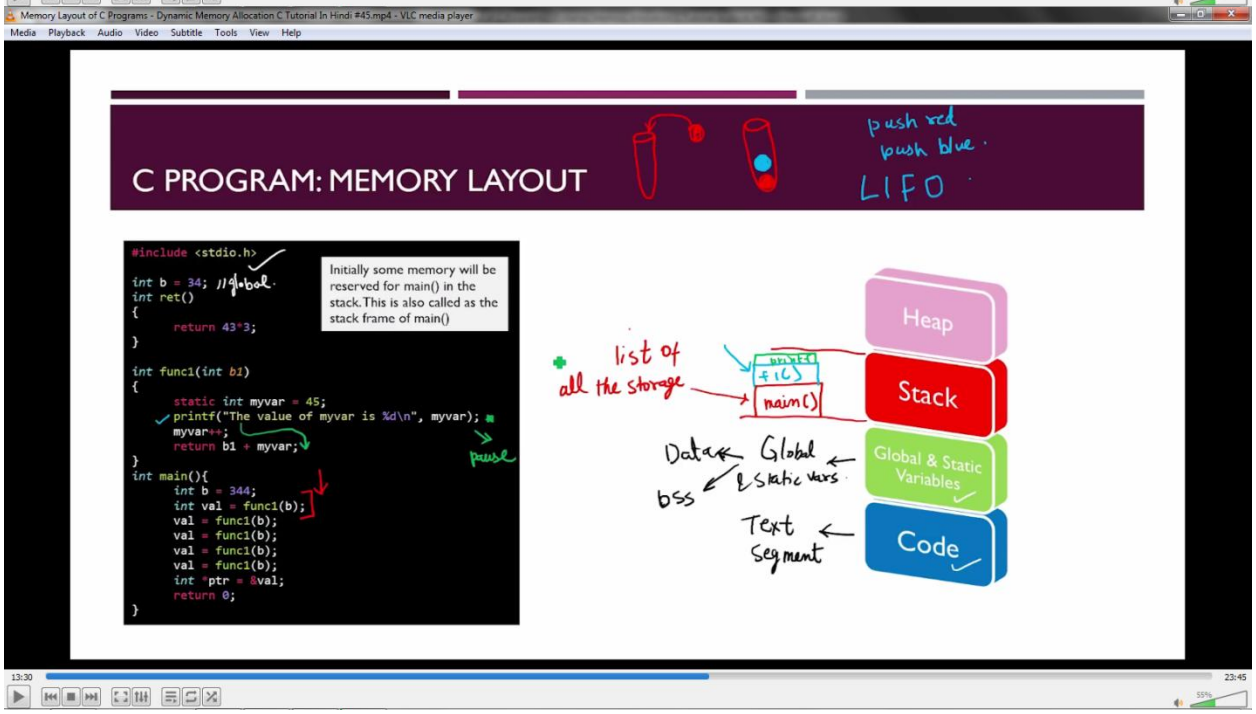

Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
WHAT & WHY DYNAMIC MEMORY ALLOCATION?
char name [39]; Shubham \0
harry \0
Shyam \0
An statically allocated variable or array has a fixed size in memory.
We have learnt to create big enough arrays to fit in our inputs but this doesn't seem like an optimal way to allocate memory.
Memory is a very useful resource.
Clearly we need a way to request memory on runtime.
Dynamic Memory Allocation is a way in which the size of a data structure can be changed during the runtime.
03:20
23:45


Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
STATIC VS DYNAMIC MEMORY ALLOCATION
int i;
Static Memory Allocation
Allocation is done before the program's execution
There is no memory reusability and the memory allocated cannot be freed.
Less efficient
Dynamic Memory Allocation
Allocation is done during the program's execution
There is memory reusability and the allocated memory can be freed when not required
More efficient
04:12
23:45





LIFO – Last in first out

Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

C PROGRAM: MEMORY LAYOUT

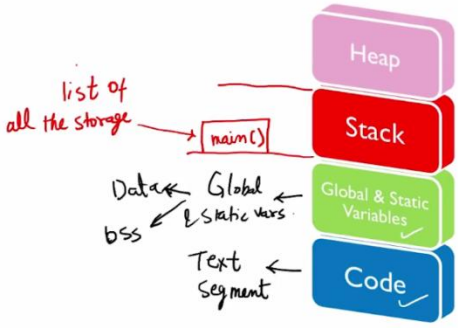
push red
push blue.
LIFO

```c
#include <stdio.h>

int b = 34; //global.
int ret()
{
      return 43*3;
}

int func1(int b1)
{
      static int myvar = 45;
      printf("The value of myvar is %d\n", myvar);
      myvar++;
      return b1 + myvar;
}
int main(){
      int b = 344;
      int val = func1(b);
      val = func1(b);
      val = func1(b);
      val = func1(b);
      val = func1(b);
      int *ptr = &val;
      return 0;
}
```

Initially some memory will be reserved for main() in the stack. This is also called as the stack frame of main()



14:12 23:45

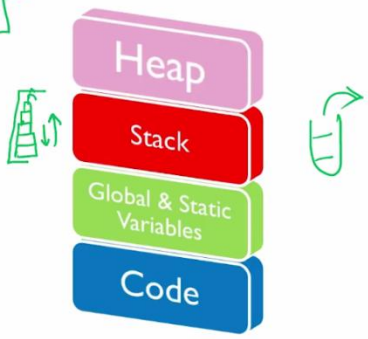
Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

C PROGRAM: STACK OVERFLOW

LIFO

- Compiler allocates some space for the stack part of the memory
- When this space gets exhausted for some bad reason, the situation is called as stack overflow
- Typical example includes recursion with wrong/no base condition.



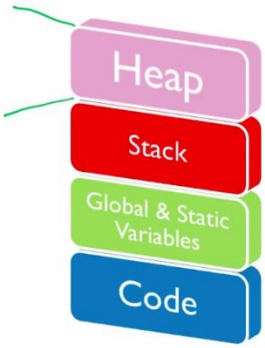
16:13 23:45

Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

C PROGRAM: USE OF HEAP

- There are a lot of limitations of stack (static memory allocation)
- Some of the examples include variable sized array, freeing memory no longer required etc.
- Heap can be used flexibly by the programmer as per his needs.



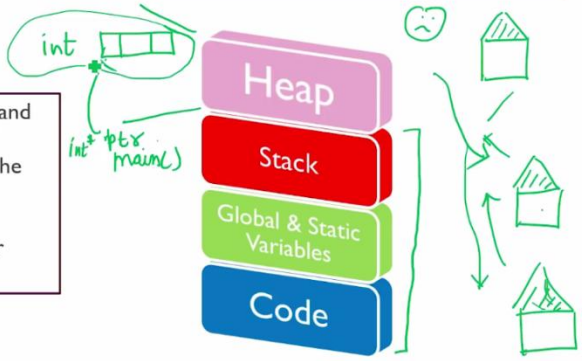
17:23 23:45

Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player

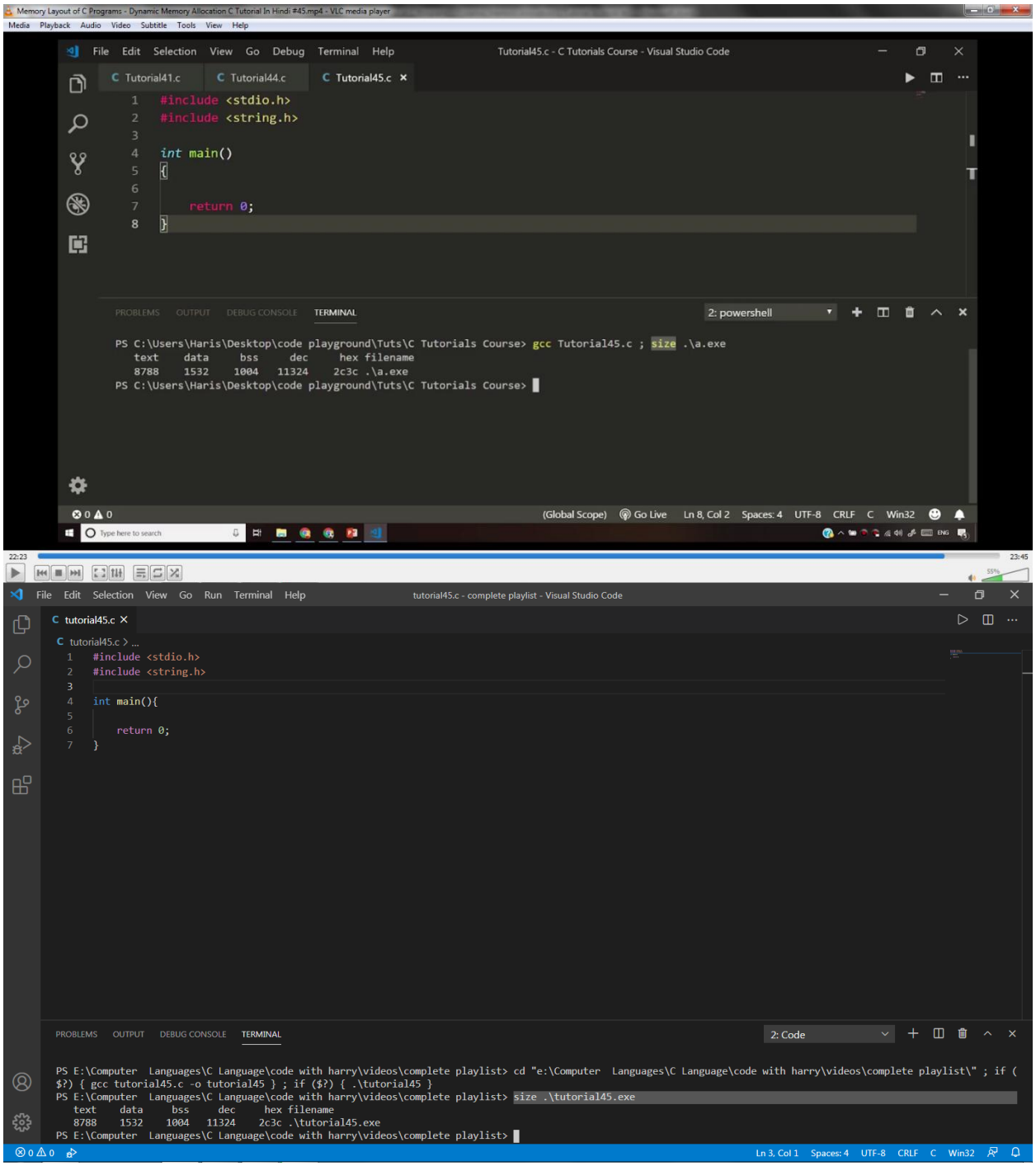
Media Playback Audio Video Subtitle Tools View Help

C PROGRAM: USE OF HEAP

- We can create a pointer in our main function and point to a memory block in the heap
- The address is stored by the local variable in the main function.
- The memory consumed will not get freed automatically in case we overwrite the pointer



20:26 23:45


Memory Layout of C Programs - Dynamic Memory Allocation C Tutorial In Hindi #45.mp4 - VLC media player
Tutorial45.c - C Tutorials Course - Visual Studio Code
Tutorial41.c
Tutorial44.c
Tutorial45.c
#include <stdio.h>
#include <string.h>
int main()
{
return 0;
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: powershell
PS C:\Users\Haris\Desktop\code playground\Tuts\C Tutorials Course> gcc Tutorial45.c ; size .\a.exe
text data bss dec hex filename
8788 1532 1004 11324 2c3c .\a.exe
PS C:\Users\Haris\Desktop\code playground\Tuts\C Tutorials Course>
tutorial45.c - complete playlist - Visual Studio Code
tutorial45.c
#include <stdio.h>
#include <string.h>
int main(){
return 0;
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Code
PS E:\Computer Languages\C Language\code with harry\videos\complete playlist> cd "e:\Computer Languages\C Language\code with harry\videos\complete playlist\" ; if ($?) { gcc tutorial45.c -o tutorial45 } ; if ($?) { .\tutorial45 }
PS E:\Computer Languages\C Language\code with harry\videos\complete playlist> size .\tutorial45.exe
text data bss dec hex filename
8788 1532 1004 11324 2c3c .\tutorial45.exe
PS E:\Computer Languages\C Language\code with harry\videos\complete playlist>