Oops Concept

Local and Global Variable

Global variables:- (1) Written Outside of a function
(2) Accessible To all func^n (samecopy)

Local variable:- (1) Written inside a func^n
(2) Accessible inside that func^n scope only
(3) Scoped.

```
#include<iostream>
using namespace std;
int x=2; // Global variable
void func(){
   int x= 60;
   cout<< x <<endl;
   ::x = 40;
   cout<< ::x<<endl;
}
int main(){
   ::x=4;    // global x
   int x= 20;  local to main func^n
   ccout<<x<<endl;
   cout<< ::x<<endl; //accessing global with ::
  {
   int x=50;
   { int x= 44;
      cout<< x<<endl;
   }
   Cout << x<<endl;
   cout<< ::x<<endl;
  }
   func();
   return 0;
```

Most recent variable scoped vise print hoga.

#output:-

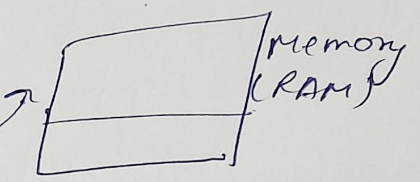
```
20
4
44
50
4
60
40
```

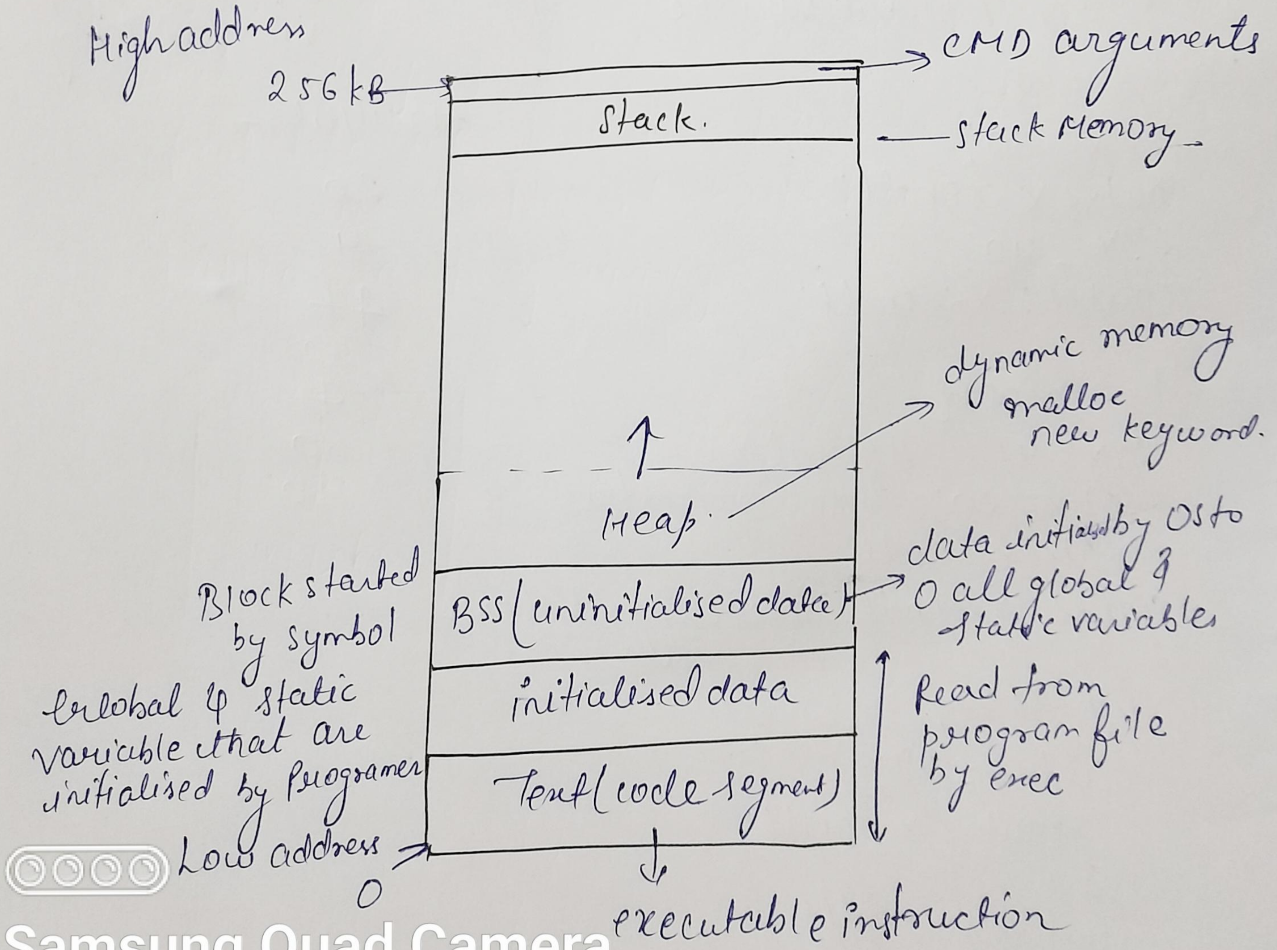
Memory Layout of a program

→ hello.cpp → compile → Assembly code/Machine code ↓ executable code ↓ "program". ↓ a.out 256kB

* compilation process required to convert the text ~~in~~ file which are present in ~~hell~~ cpp file into the program.

exec → kernel → Program




High address
256kB
CMD arguments
Stack.
stack Memory.
dynamic memory
malloc
new keyword.
Heap.
Block started by symbol
BSS (uninitialised data)
data initialised by OS to 0 all global & static variables
Global & static variable that are initialised by Programer
initialised data
Read from program file by exec
Text (code segment)
Low address
0
executable instruction