Lecture - 27

functions in C - 05

↳ Scope of a Variable &
Storage Classes

⇒ Programming in C

```
Void A ( ) {
    int x = 10;
   int m = B(x) + 10;
    printf (" %d", m);
}

int B ( int a ) {
     return (a + 100);
 }

int main ( ) {
 int p = 50;
   A ( );
  printf (" Successful");
  return 0;
 }
```

m = 120

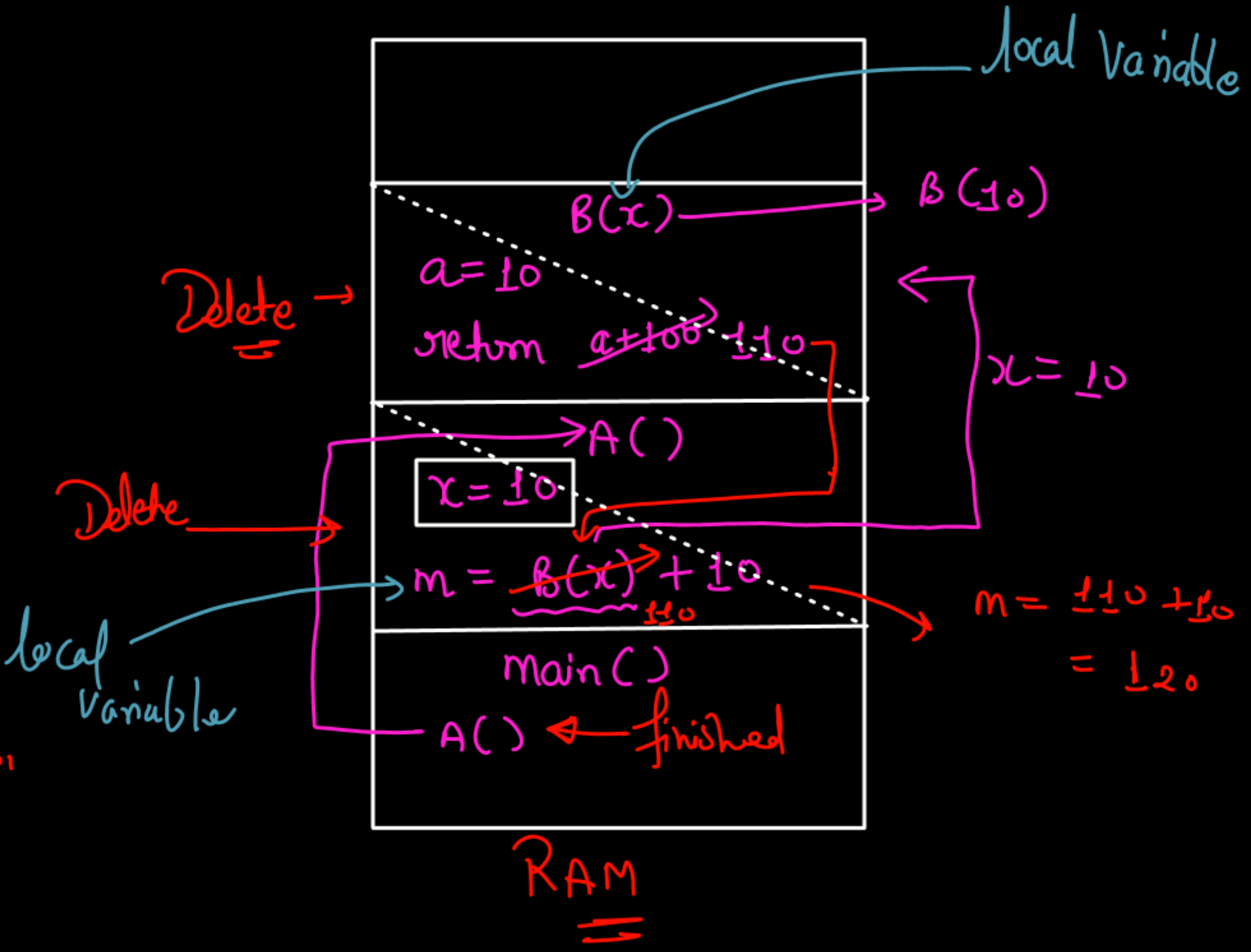
'120'

entry Point

local Variable for main ( )

"Succesful"

* When we call a function, then the Activation record of that function is generated inside RAM.

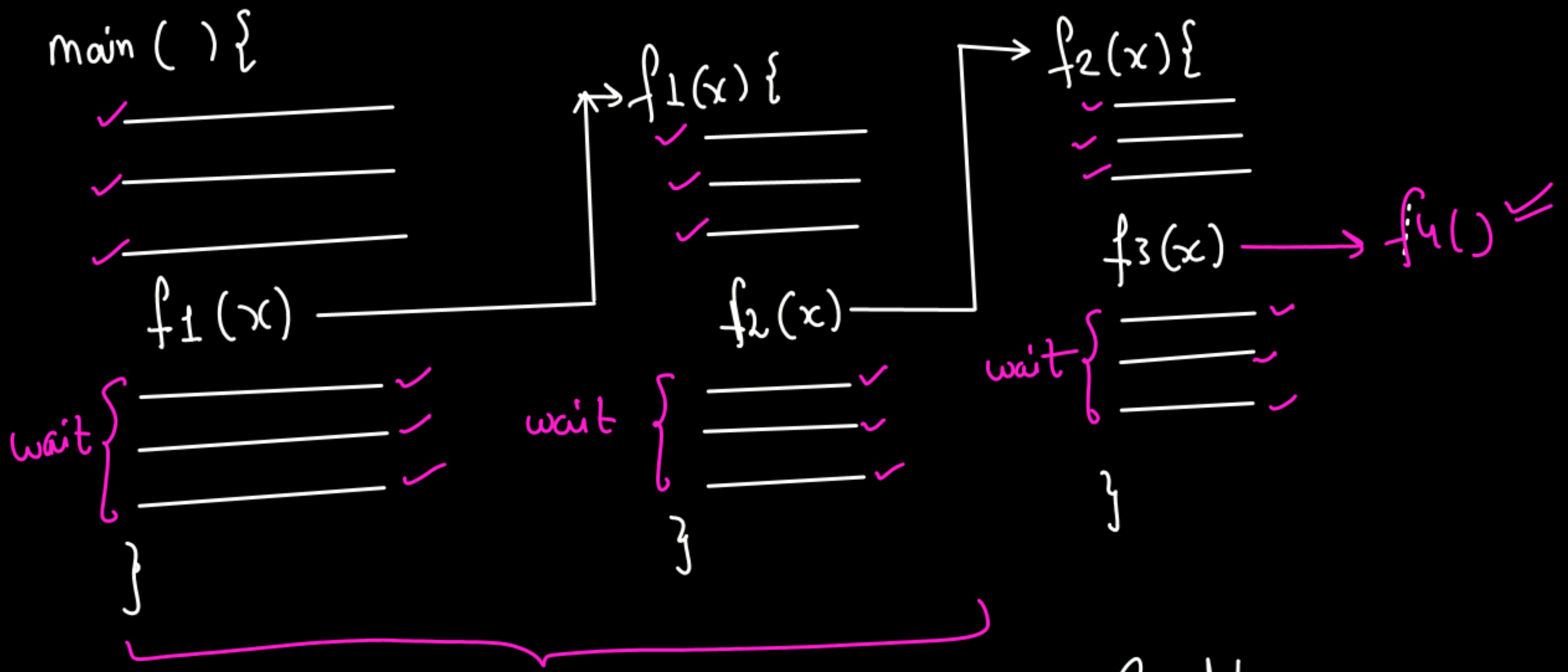


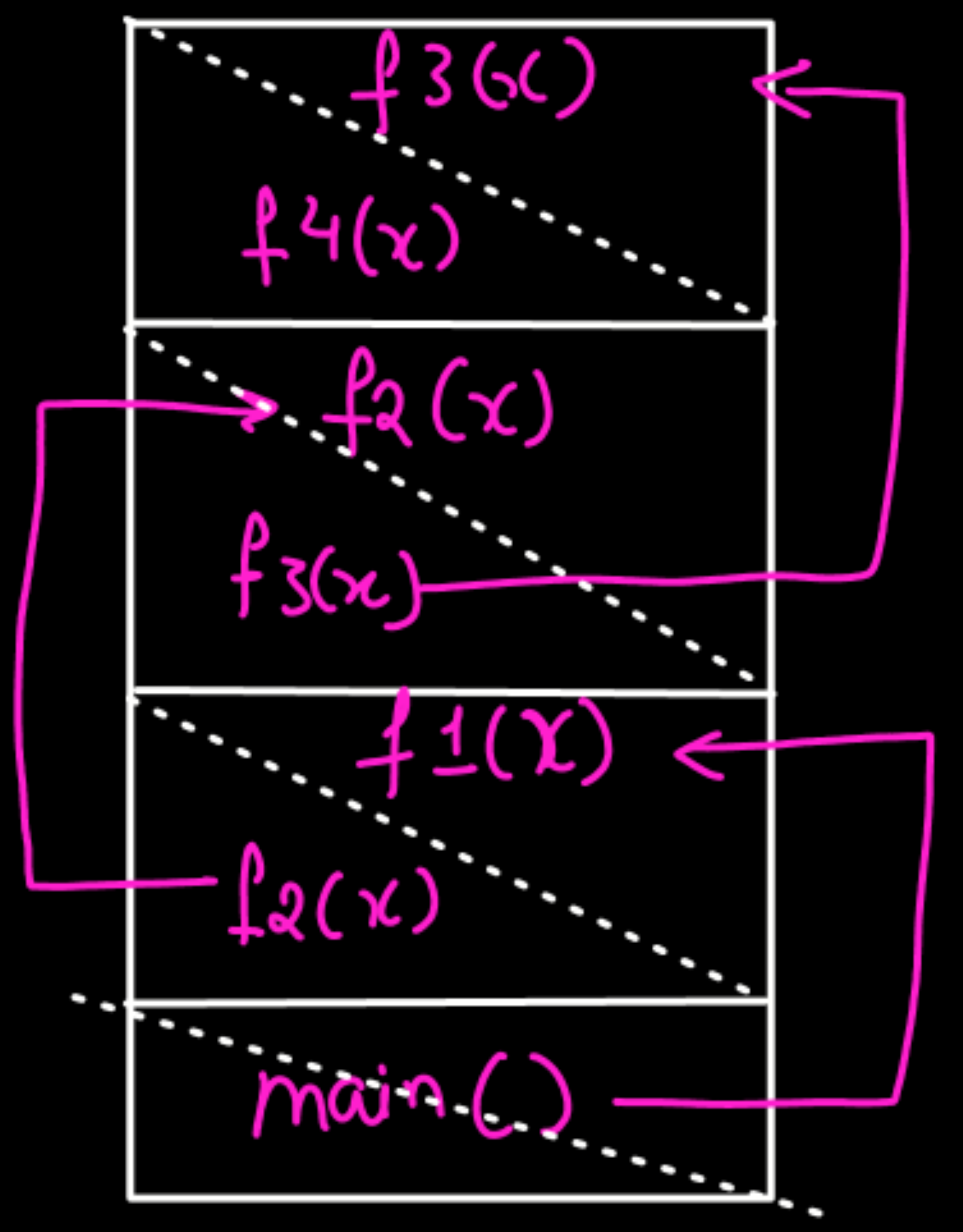
Whenever the function finish its all lines of Code, then the Activation Record will be deleted.

↳ When AR Cleared, the variables, objects or other function element will also be destroyed.

⇒ The lifetime of a function in a Memory is dependent on their return value.

⇒ That means the variables defined inside the function or function parameters have their local scope.





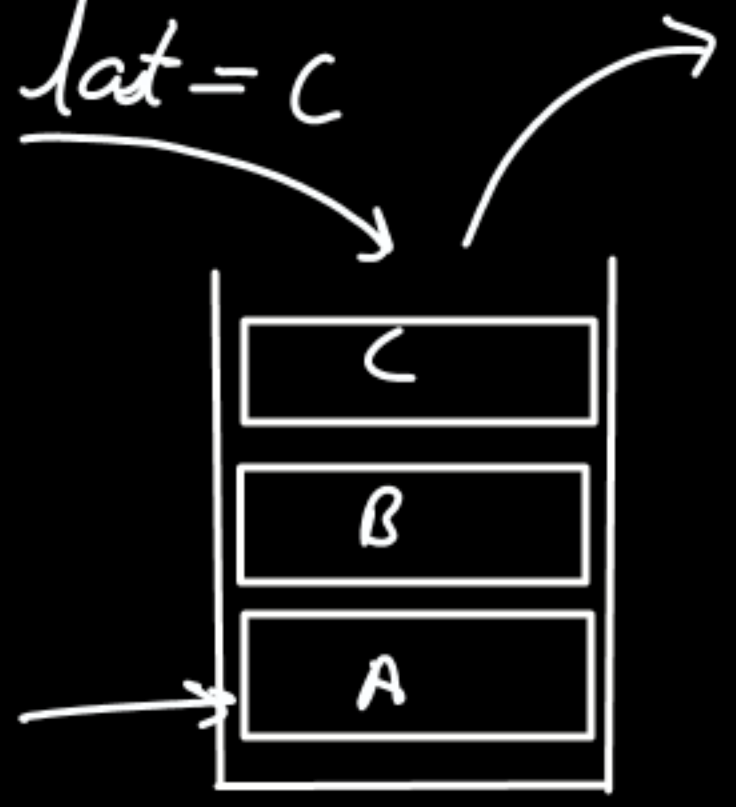
Runtime Stack

Concept

↳ Approach that use last in first out policy

LIFO

Chapter ← [Data Structure]



main( ) ⟶ f1(x) ⟶ f2( ) ⟶ f3( ) ⟶ f4( )

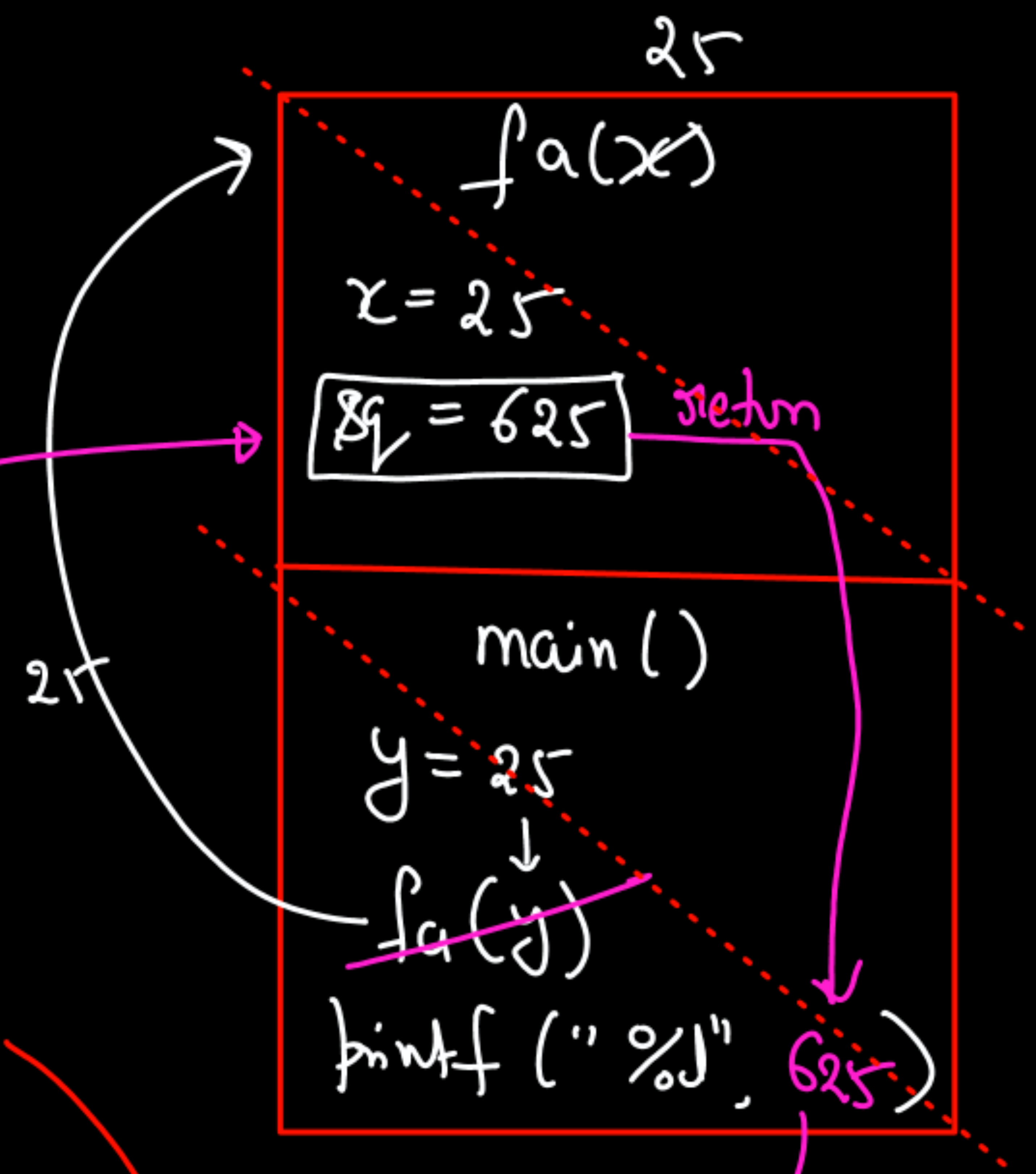
```
# include <stdio.h>
Void  fa ( int x) {
  int sq =  x * x ;
     return sq;
}
int p = 100;
int main ( ) {
   int y = 25;
   printf ("%d", fa(y));
  return 0;
}
```

allowed

global Scope

625

"625"



local Variables

- → main ( ) → y
- → fa () → x, sq

} lifetime ⇓ (with function Execution)

* The Variables that are not the part of a function are termed as global Variable.

⇒ Scope & lifetime = During the program.

Local Variable:

↳ The variables that are the part of a function.

↳ defined, declared, function parameters

→ lifetime → during function execution.

Global Variable:

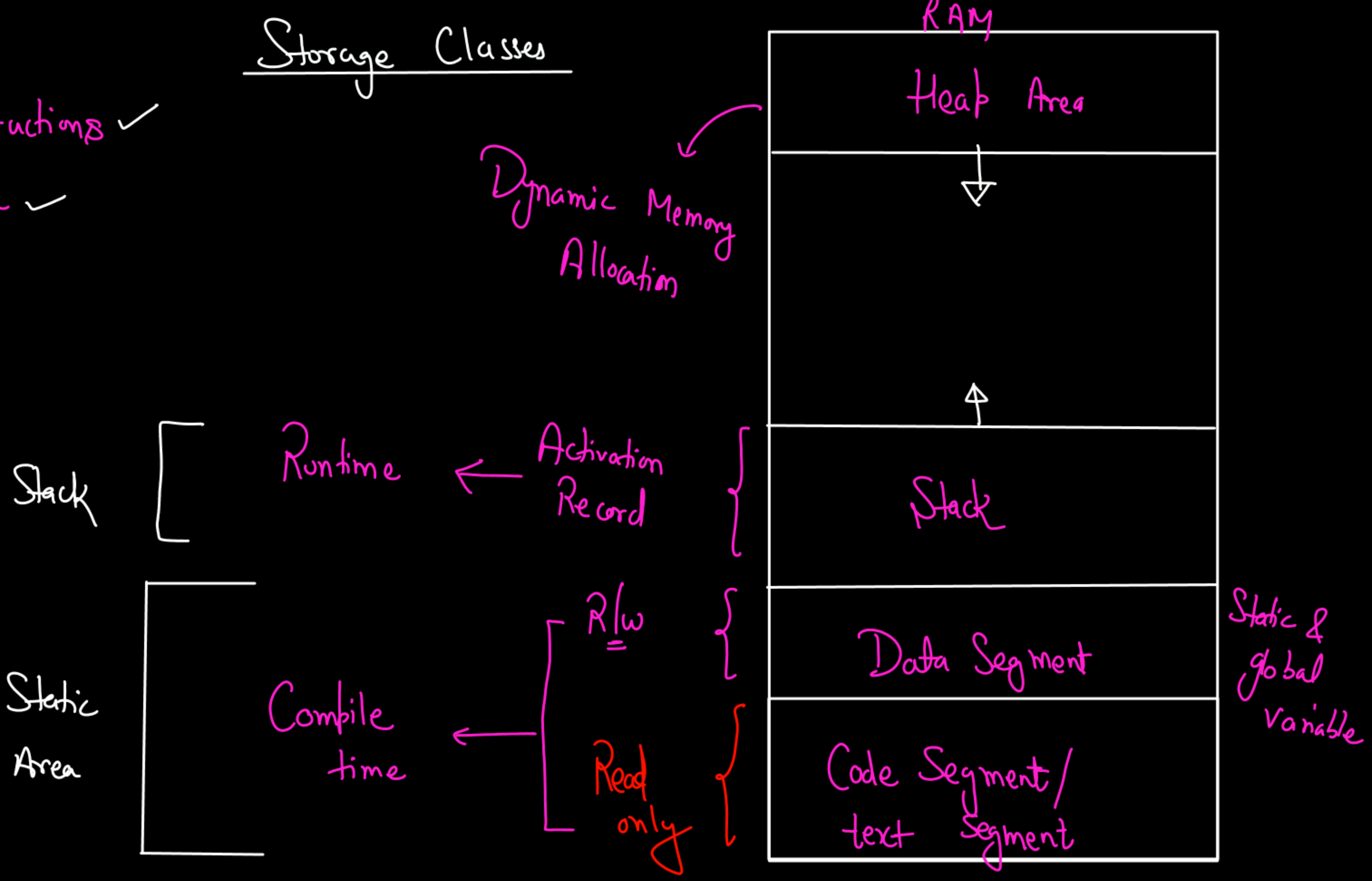
↳ Not the part of any function.

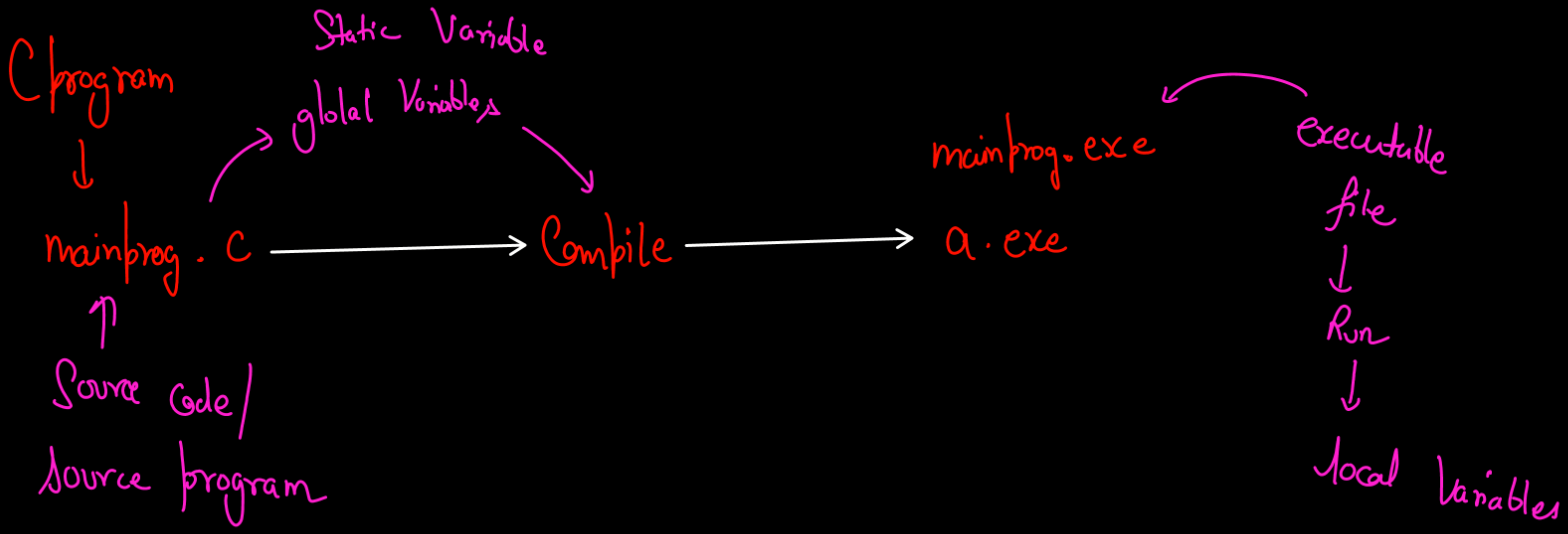
↳ Can be accessed by outsiders.

↳ lifetime → during whole program

Storage Classes

Program → Instructions ✓
Program → Data ✓




C program
mainprog . c
Source Code/
Source program
Static Variable
global Variables
Compile
a . exe
mainprog. exe
executable
file
Run
local Variables

tomorrow →

⇒ Storage Classes

↳ Variables → Macros → global → Register