(Video 12) Scope of Variables → local vs Global.

Outline

→ scope of variable

→ Global variable

→ Local variable

$\boxed{\text{Scope = lifetime}}$

The area under which a variable is applicable.

→ a block or region where a variable is declared, defined and used and when a block ~~is~~ or a region ends, variable is automatically distroyed.

Code 1

```
# include <stdio.h>
int main() {
    int var = 34;
    printf("%d", var);
    return 0;
}.
int fun() {
    printf("%d", var);
}
```

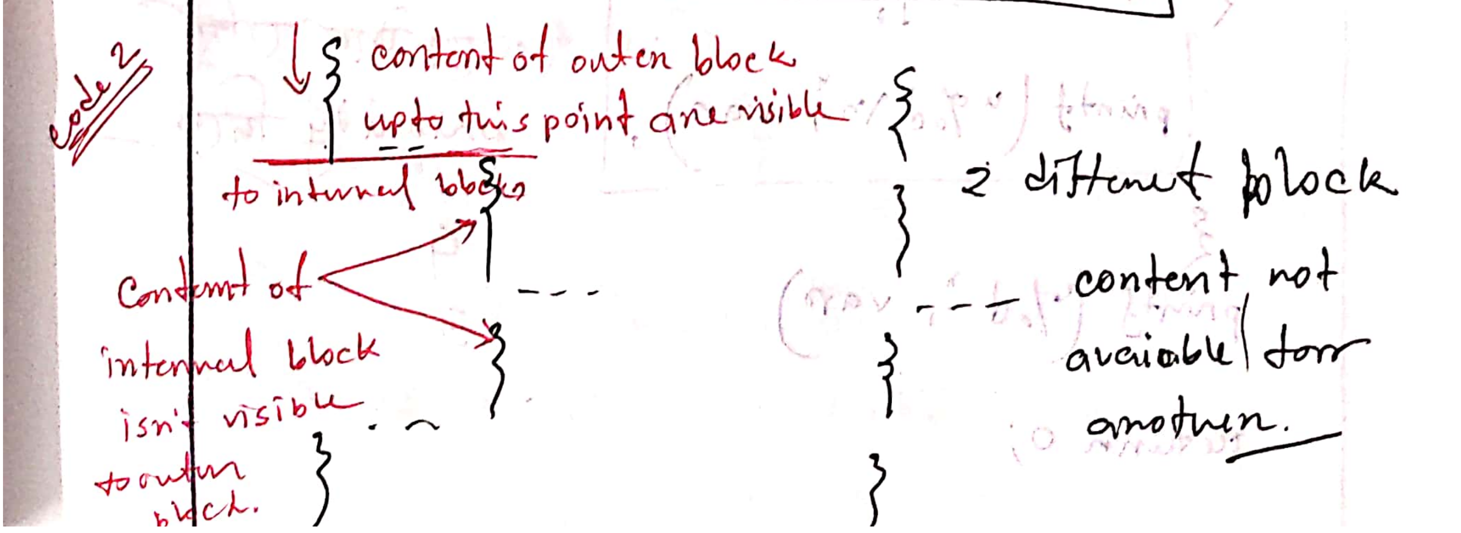
scope of this variable within main function only. Therefore called LOCAL to main() function.

Trying to access variable "var" outside main function.

Therefore, it will return Error

var not declared

Code 2


content of outer block upto this point are visible to internal blocks

Content of internal block isn't visible to outer block.

2 different block content not available for another.

Code 3

```
#include <stdio.h>

int main() {
    int var = 3;
    int var = 4;

    printf ("%d \n", var);

    printf ("%d", var);

    return 0;

}.
```

একই নামে variable দুই বার decline করছি।

Error আসবে, redefinition of "var"

Code 4

```
#include <stdio.h>

int main() {

    int var = 3;

    { int var = 4;

    printf ("%d \n", var);

    }
    printf ("%d", var)

    return 0;
```

~~4:3~~

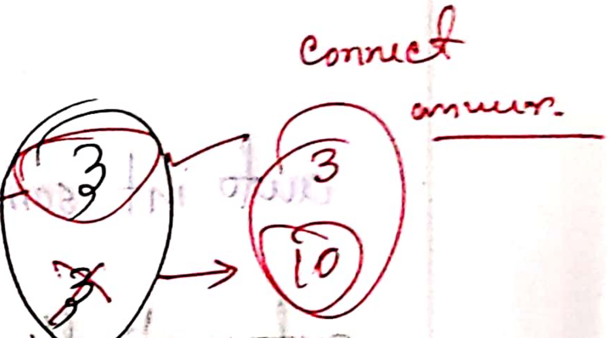
4
3

ভেতরের টাকে বেশি priority দিবে।

4
3

Code 5

```
#include<stdio.h>
~~int main()~~ {
int fun();
int van=10;   → Global Variable
int main() {
    int van = 3;
    printf("%d\n", van);
    fun();   → calling function
    return 0;
}
int fun()
{
    printf("%d", van);
}.
```


Connect answer.
3
~~3~~
3
10