Defining Scope

Scope = Lifetime

The area under which a variable is applicable or alive.

Strict Definition: a block or a region where a variable is declared, defined and used and when a block or a region ends, variable is automatically destroyed.

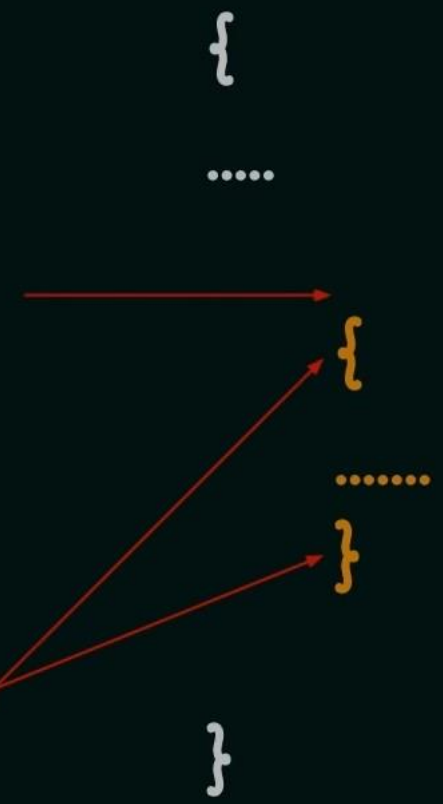
```c
#include <stdio.h>

int main() {
    int var = 34;
    printf("%d", var);
    return 0;
}

int fun()
{
    printf("%d", var);
}
```

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

```
error: 'var' undeclared (first use in this function)
```

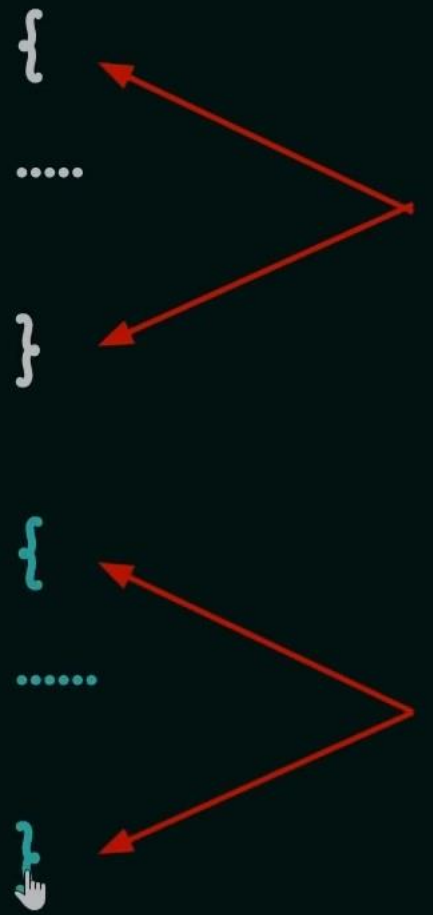
Trying to access variable 'var' outside main() function

Basic Principle of scoping



Contents of outer block upto this point are visible to the internal block

Contents of internal block are not visible to outer block.



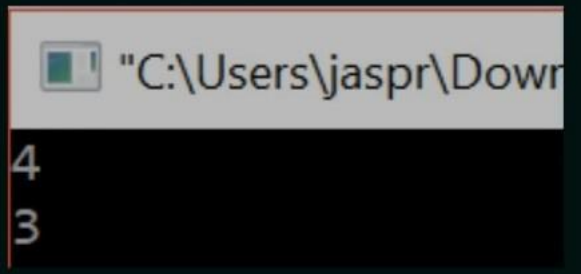
Contents of this block is not visible to any block outside to this block.

Contents of this block is not visible to any block outside to this block.

```c
#include <stdio.h>

int main() {
    int var = 3;
    int var = 4;
    printf("%d\n", var);
    printf("%d", var);
    return 0;
}
```

```
error: redefinition of 'var'
```

```c
#include <stdio.h>

int main() {
    int var = 3;
    {
        int var = 4;
        printf("%d\n", var);
    }
    printf("%d", var);
    return 0;
}
```



```c
#include <stdio.h>
int fun();

int var = 10;

int main() {
    int var = 3;
    printf("%d\n", var);
    fun();
    return 0;
}

int fun()
{
    printf("%d", var);
}
```

This variable is outside of all functions.
Therefore called a GLOBAL variable

Output: 3

Output: 10

It will access the GLOBAL variable.