

WHAT IS AUTO MODIFIER?

Auto means Automatic

Variables declared inside a scope by default are automatic variables.

Syntax: `auto int some_variable_name;`

```
#include <stdio.h>

int main() {
    int var;
    return 0;
}
```

==

```
#include <stdio.h>

int main() {
    auto int var;
    return 0;
}
```

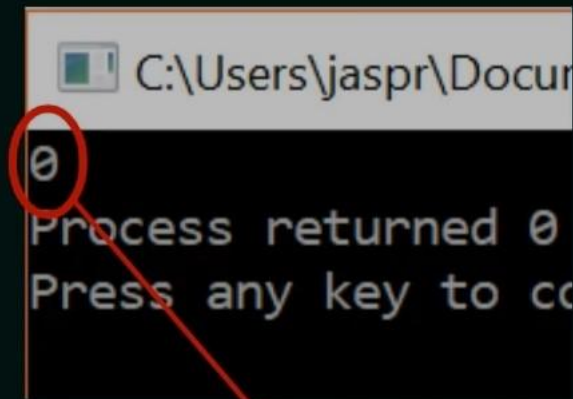
TAKE AWAYS

2. On the other hand, global variable by default initialized to 0.

Global
Variable

```
#include <stdio.h>

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



As expected

EXTERN MODIFIER

`int var;`



Declaration and Definition

`extern int var;`



Declaration

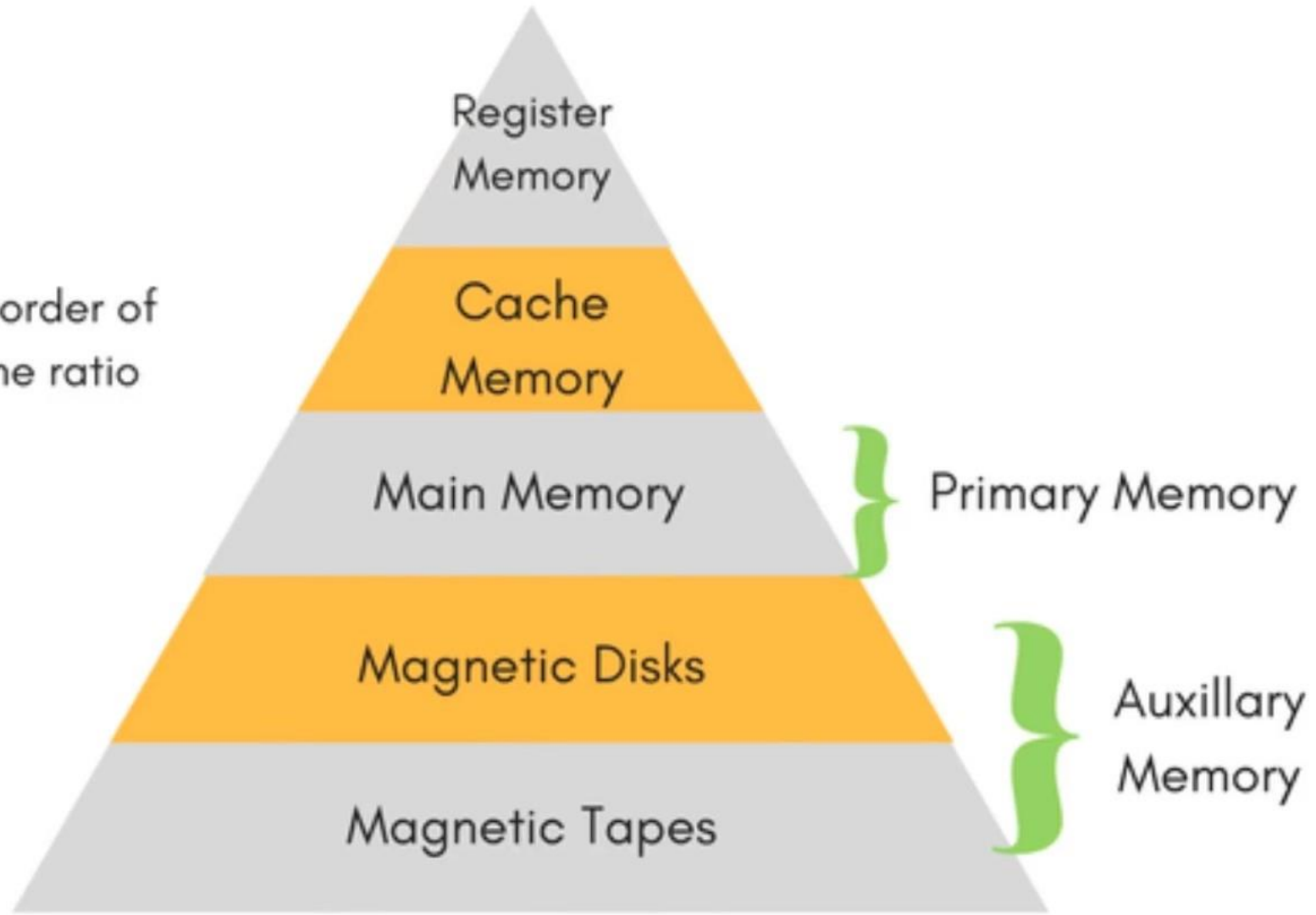
- Extern is short name for external.
- Used when a particular file needs to access a variable from another file.

TAKE AWAYS

1. When we write `extern some_data_type some_variable_name;` no memory is allocated. Only property of variable is announced.
2. Multiple declarations of extern variable is allowed within the file. This is not the case with automatic variables.
3. Extern variable says to compiler “go outside from my scope and you will find the definition of the variable that I declared”.
4. Compiler believes that whatever the extern variable said is true and produce no error. Linker throws an error when he finds no such variable exist.
5. When an extern variable is `initialized`, then memory for this variable is allocated and it will be considered `defined`.



Increasing order of
access time ratio



WHAT IS REGISTER MODIFIER?

Syntax: register some_data_type some_variable_name

```
#include <stdio.h>

int main() {
    register int var;
    return 0;
}
```

WHAT IS REGISTER MODIFIER?

Register keyword hints the compiler to store a variable in register memory.

This is done because access time reduces greatly for most frequently referred variables

This is the choice of compiler whether it puts the given variable in register or not.

Usually compiler themselves do the necessary optimizations

TAKEAWAYS

1. Static variable remains in memory even if it is declared within a block on the other hand automatic variable is destroyed after the completion of function in which it was declared.
2. Static variable if declared outside the scope of any function will act like global variable but only within the file in which it is declared.
3. You can only assign a constant literal (or value) to a static variable.