DAY # 27

Monday, June 22, 2026 8:34 PM

extern storage class in C

extern ⇒ is keyword
⇒ It is used to declare a variable which is defined somewhere else
⇒ Memory is not allocated for these type of variables.
⇒ It reference to variable defined at some other place or in any header file

Syntax:- extern int a;

int a; ⇒ Definition (We are defining a variable with datatype int + Memory is allocated.
int a = 5; ⇒ Definition + initialization (Memory allocation + initialisation)

extern int a; ⇒ Declaration of a (No memory allocation)
⇒ a is defined somewhere else

Example 1:-

```
#include <stdio.h>
int a = 100;          <- Definition / Memory (4 Byte)

int main()
{
    extern int a;     <- Declaration
    printf("%d", a);  <- 100
    return 0;
}
```

Example 2:-
1) We will create our own header file.
2) We will define variable 'a' in the header instead of defining a above main() function

How to create header file?

myheader.h // Any name with .h extention.

```
int a = 100
```

How to use / include in our program?

```
#include<stdio.h>
#include "myheader.h"      // int a = 100;

int main()
{
    extern int a;
    printf("%d", a)

    return 0;
}
```

Scope of variables

Example1:-

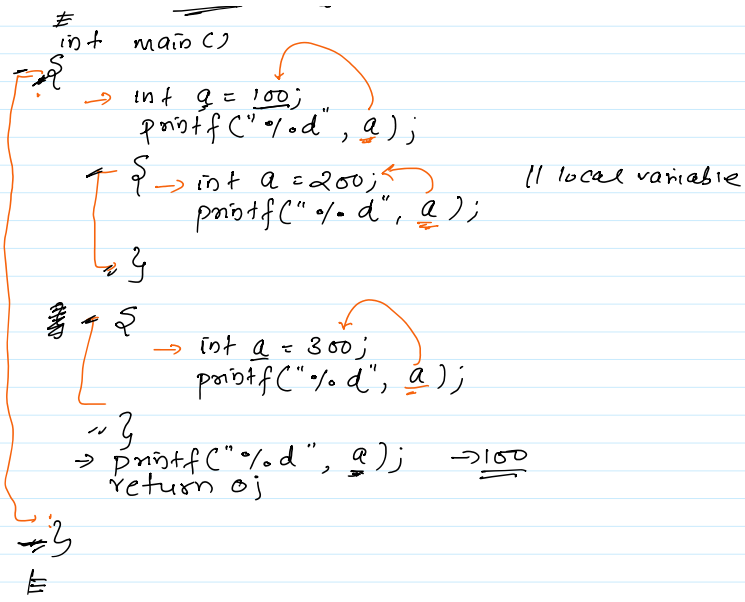
```
int main()
{
    int a = 100;
    printf("%d", a);        -> 100
}
```

≡

```
int a = 1000;   ?
int main()
{
    int a = 100;
}
```

Example 2:-

Example 2:-



Example 3:-

```
#include <stdio.h>
int a;
int main()
{
    auto int b;
    return 0;
}
```

Global variables ⇒ Default value is 0

// local variable ⇒ Default is Garbage/indeterminate

// a will always print 0
// b may print 0 sometimes but we should say it will print garbage indeterminate