(Video 13) Variable Modifiers - Auto and Extern.

Auto Modifier ~~#~~ **auto variable**

Variable decleared inside a scope by ~~default~~ default are automatic variable

```
auto int some_variable_name;
```

automatically get destroyed after the completion of function

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

=

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

main function শেষ হবে varr ও শেষ।

memory

automatic variable ~~an~~ won't waste memory. Function শেষ হলে memory reallocate করে দেয়

→ if we don't initialize auto variable, by default it will be initialized with some garbage (random) value.

```c
#include<stdio.h>
int main()
{ auto int var;
  printf("%d \n", var);
  return 0;
}.
```

→ তোমার random value print করবে

like python random.

→ Global variable by default int initialized to zero. (0)

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

int var; → default value zero ধরে নিবে

Extern Modifier:

int var

↓

Declaration + Definition

↓ variable এর type specify করে দেয়া।

Integer বা Character Etc.

↘ variable এর জন্য memory allocate করা। By default 0 allocate করা। But যদি var = 9 এভাবে বলা হয় তাহলে initialization বলা।

Ex extern int var;

↓

Declaration

extern use করলে কম্পিউটার declare করবে।

extra কোনো memory allocate করবে না।

mainly ~~we~~ what is happening here is,

extern int var; এখানে, var নামের variable এর জন্য কোনো memory allocate করবে না। শুধু define করবে না। Just declear করবে।

Code external source থেকে খুঁজে দেখবে এই variable টি কোথায় আছে। তার code থেকে তাকে import করবে।

→ extern is short name for external.

→ used when a particular file needs to access a variable from another file.

Bothe code same project এ থাকবে।

Same project.

main.c

```
#include <stdio.h>
~~int main()~~
extern int a;
int main() {
    printf("%d", a)
    return 0;
}
```

Output → 9

other.c

```
int a=9
```

(i) same project এ
থাকতে হবে same
থাকলে ?

(ii) same code

{ } declare

{ } use

কাজের extern
use করবে

(ii) { } declare
{ } use

main.c → extern use
না করলে output
আসবে 0.

Code runs through
Linker

code combine করে run করবে

→ If we write

```
extern int a;
extern int a;
extern int a;
```

} multiple time in main.c

No Error.

Because memory তিতে কোনো জায়গা খরচা নাই. just declare করতেছে.

→ ~~main~~ main.c তে যদি,

```
int a=19
int main() {
    extern int a;
    printf("%d", a);
    return 0;
}
```

other.c তে যাবে না আগে code এ মধ্যে খুঁজে দেখবে. Then ~~[illegible]~~ বাইরে যাবে.

~~Output → 19~~

এখানে যদি

extern int a = 5; লিখলে

তখন declare & define both করবে।

এবং so memory allocation হবে।

Summary:

→ When we write,

[extern v-type v-name;]

no memory allocated. Only property announced.

→ multiple declaration of extern variable allowed.

→ It says "Go outside my scope and find the definition of variable that I declared.

→ Compiler delays extern variable

→ When extern variable initialized, then memory for this variable to allocated. Defined