(Video 15) Variable Modifiers - Static

→ make ~~an~~ a project / Folder

main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int value; = 0
(*1) value = increment();
    value = incremnt ();
    value = increment ();

printf ("%d", value);
    return 0;

}.
```

add.c

```c
int increment() {
    int count = 0;   // 1
    count = count + 1;
    return count;

}.
```

Output → 1
But it should be 3

Step1: increment() function টা run করানোর পর count এর value 1 return করাচ্ছে।
তার জন্য main function এর (*1) এ value = 1 হচ্ছে।

Step 2: এর পরে increment() function আবার শুরু থেকে শুরু হবে। count এর যে আগের value ছিল ওগুলোর existence নাই।

So, আবার count এর value 1 return করবে।

value = 1 হবে।

3rd sept: Same, at the end of code it'll return

value = 1.

So overall output = 1

So, ultimately increment হচ্ছে না।

এখন Increment's function কে modify করলাম।

add.c

```
int count = 0
int increment() {
    count = count + 1;
    return count;
}
```

এখানে, `int count = 0` Globally declare করায় এটা অন্য নির্দিষ্ট কোনো Function এর ভিতরে নাই, ফলে 2nd time count এর value হারায় যাচ্ছে না। (1) থেকে শুরু হচ্ছে।

Output will be, 3

1st value = 1

2nd value = 2

3rd value = 3

→ এখানে problem যেটা হচ্ছে, এই Global variable টা পুরো project এর মধ্যে available.

main.c   add.c উভয়েই এটার access করতে পারবে।

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
extern int count;

int main() {
  int value;
  value = increment();
  value = increment();
  value = increment();
  = count = count + 3;
  value = count;
  printf("%d", value);
  return 0;
}
```

```c
int count = 0;
int increment() {
  count = count + 1
  return count
}
```

Not Good Practice

value = increment(); → value = 3 এই position এ.

extern command এর কাজ

add.c থেকে count এর value call করছে।

3+3 = 6

Output → 6

এখানেই main ~~এর মধ্যে~~ ডাকা টা হচ্ছে।

int count ~~= 0~~; in (add.c)

এটা তোমার project এ visible. যে কেউ use করতে পারবে।

But এমন হতে পারে যে, তুমি এটা section এ Globally Define করলাম, but ~~এর~~ project এর অন্য public না, যেন এটা access না করতে পারে।

Static int count ~~= 0~~

এখন আর extern int count; এই command use করে আর কেউ এই count access করতে পারবে না।

now, count not visible outside ~~main.c~~ (add.c)

~~#in~~ inc

```
int increment() {
    static int count;
    count = count + 1;
    return count;
}
```

এবার এখন main. C চালালে output 1 আসবে না

~~Ex~~ output → 3

কারণ static ডেক্লেয়ার করলে তা বার বার zero নিবে না previous value টাই নিয়ে যাবে।

এটাই advantage of static

Step - 1 → 1

" 2 → 2

" 3 = 3

→ Static variable remains in memory even if it is declared within a blocks. - on the other hand automatic variable always distroyed after the completion of the function

→ 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.

→ we can assign constant in static value but not through variable

```
static var = int var = 3
```

Right ✓

```
int varr = 3
static int varr2 = varr
```

Wrong ✗