(Video 26) Logical Operators in C.

&& → and

|| → or

! → not.

→ && and || are used to combine two conditions

(AND) (OR)

for && return True if all True.

any one or more than one False it will return false.

1 1 = 1

0 1 = 0

1 0 = 0

0 0 = 0

```
int a = 5;
if ( a==5 && a != 6 && a <= 56 && a > 4)
{
    printf ("Welcome!");
}
```

For || operator, True when one or more than one is ~~true~~ True. and ~~returns~~ False when all conditions are false.

0 0 → 0

0 1 → 1

1 0 → 1

1 1 → 1

(a != 5 || a == 6 || a > 56 || a > 4)

এখানে 3টা False হলেও last টা True হওয়ায় call True.

For ! (NOT) operator

- returns true when ~~sad~~ condition false

" False " " " TRUE.

```
int a=5;
if(!(a==6))
{
printf("Conditional Operator");
}
```

Short Circuit in case of &&: if there is a condition anywhere in the expression that returns false, then the rest of the conditions after that will not be evaluated.

```
#include<stdio.h>
int main() {
  int a = 5, b = 3;
  int incr;
  incr = (a<b) && (b++);
  printf("%d \n", incr);
  printf("%d", b);
  return 0;
}
```

output
0
3

False = 0
True = 1

(a<b) is ①, (b++) is ②

→ a = 5, b = 3 so (a<b) condition false

So, it won't check any others condition, instade will return false directly.

incr will be printed as 0 (False)

and post increment condition not worked, so, b will print 3

* in ~~and or~~ && or || condition

condition && condition
condition || condition
} usual structure

But if,

condition && Expression
condition || Expression
} IF condition is filled up the expression will count as = TRUE

```
#include<stdio.h>
int main(){
  int a=5, b=3;
  int incr;
  incr = (a>b) && (b++);
  printf("%d\n", incr);
  printf("%d", b);
  return 0;
}
```

(a>b) → condition TRUE
(b++) → This will also count as True
incr will print 1
b will print 4

Short Circuit in Case of ||: if there is a condition anywhere in the expression that returns TRUE, then the rest of the conditions after that will not be evaluated.

```c
#include <stdio.h>
int main () {
    int a = 5, b = 3;
    int incr;
    incr = (a > b) || (b++);
    printf ("%d\n", incr);
    printf ("%d", b);
    return 0;
}.
```

যে কোনো একটা TRUE হলেই আর বাকি গুলোতে যাবে না।

(a > b) → TRUE

(b++) → এটাতে আর যাবেই না।

→ incr will return TRUE printed as 1.

→ b printed as 3

কারণ b++ পর্যন্ত যায় নি।