Video No.: 32

Topic: Conditional Operator

$\boxed{\text{Condition ? if TRUE : if}}$

>> Understanding of Conditional Operator:

```
char result;
int marks;
if (marks > 33){
    results = 'p';
}else{
    results = 'f'
}
```

```
char result;
int marks;
results = (marks > 33) ? 'p' : 'f';
```

```
Results =     (marks > 33)    ?    'p'     :     'f';
               Condition          TURE           FALSE
```

>> This is a conditional expression, which is after all an expression, therefore it is an rvalue and the results is lvalue.

>> Conditional Operator is the only **Ternary Operator** available in the list of operators in C language.

>> As in Expression_1 ? Expression_2 : Expression_3;

Expression_1 is the Boolean expression. IF we simply write 0 instead of some Boolean expression then that simply means FASLE and thus Expression_3 will get evaluated.

Question: What will be the output?

```
#include <stdio.h>

int main()

{

    int var = 75;

    int var2 = 56;

    int num;

    num = sizeof(var) ? (var2 > 23 ? ((var == 75) ? 'A' : 0) : 0) : 0;

    printf("num = %d", num);

    return 0;

}
```

Solution:

let’s break the problem into part

(1) At first the condition is sizeof(var)

if this condition is evaluated to be true then (var2 > 23 ? ((var == 75) ? 'A' : 0) : 0) will be returned.

if false then 0 will be returned.

As we know sizeof() is an unary operator which returns how many byte a datatype can hold as var is an variable of integer data type , sizeof(var) will either return 2 or 4 as machine to machine int vary . We know every number except 0 is evaluated to be true. So (var2>23 ? ((var==75) ? 'A' : 0) : 0) it will returned.

(2) Next the condition is (var2>23)

if this condition is evaluated to be true then ((var == 75) ? 'A' : 0) will be returned

if false then 0 will be returned

as we know 56>23 is true then ((var==75) ? 'A' : 0) will be returned.

(3) The condition is (var == 75)

if this condition is evaluated to be true then 'A' will be returned

if false then 0 will be returned

As 75 == 75 then 'A' will be returned and stored into num variable

As c support auto type casting so int can store char.

In the final printf function we use the placeholder %d and it print integer value.

According to Ascii integer value of 'A' is 65

So, the output will be 65.