

Logical Operators:

These operators are used to combine the results of two or more conditions. An expression containing logical operator returns either 0 or 1 depending upon whether expression results true or false. Logical operators are commonly used in [decision making in C programming](#).

Operator	Meaning	Example	Return value
&&	Logical AND	(9>2)&&(17>2)	1
	Logical OR	(9>2) (17 == 7)	1
!	Logical NOT	29!=29	0

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false.
	Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	(A B) is true.
!	Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	!(A && B) is true.

Example:

```
#include <stdio.h>

main() {

    int a = 5;
    int b = 20;
    int c ;

    if ( a && b ) {
        printf("Line 1 - Condition is true\n" );
    }

    if ( a || b ) {
        printf("Line 2 - Condition is true\n" );
    }

    /* lets change the value of a and b */
    a = 0;
    b = 10;

    if ( a && b ) {
        printf("Line 3 - Condition is true\n" );
    } else {
```

```

printf("Line 3 - Condition is not true\n" );
}

if ( !(a && b) ) {
printf("Line 4 - Condition is true\n" );
}
}

```

When you compile and execute the above program, it produces the following result –

```

Line 1 - Condition is true
Line 2 - Condition is true
Line 3 - Condition is not true

```

Conditional Operator/ Ternary operator:

conditional operator checks the condition and executes the statement depending of the condition. A conditional operator is a ternary operator, that is, it works on 3 operands. Conditional operator consist of two symbols.

- 1.Question mark ?
- 2.Colon :

Syntax : condition ? exp1 : exp2;

It first evaluate the condition, if it is true (non-zero) then the “exp1” isevaluated, if the condition is false (zero) then the “exp2” is evaluated.

```

#include <stdio.h>
int main(){
char February;
int days;
printf("If this year is leap year, enter 1. If not enter any integer: ");
scanf("%c",&February);
// If test condition (February == '1') is true, days equal to 29.
// If test condition (February =='1') is false, days equal to 28.
days = (February == '1') ? 29 : 28;
printf("Number of days in February = %d",days);
return 0;
}

```

Output

```

If this year is leap year, enter 1. If not enter any integer: 1
Number of days in February = 29

```

Switch statement : when there are several options and we have to choose only one option from the available ones, we can use switch statement. Depending on the selected option, a particular task can be performed. A task represents one or more statements.

Syntax:

```
switch(expression)
{
  case value-1:
    statement/block-1;
    break;
  case value-2:
    statement/block t-2;
    break;
  case value-3:
    statement/block -3;
    break;
  case value-4:
    statement/block -4;
    break;
  default:

  default- statement/block ;
  break;
}
```

The expression following the keyword **switch** in any „C“ expression that must yield an integer value. It must be an integer constants like 1,2,3 .

The keyword **case** is followed by an integer or a character constant, each constant in each must be different from all the other.

First the integer expression following the keyword **switch** is evaluated. The value it gives is searched against the constant values that follow the **case** statements. When a match is found, the program executes the statements following the case. If no match is found with any of the case statements, then the statements following the **default** are executed.

Rules for writing switch() statement.

- 1 : The expression in switch statement must be an integer value or a character constant.
- 2 : No real numbers are used in an expression.
- 3 : The default is optional and can be placed anywhere, but usually placed at end.
- 4 : The case keyword must terminate with colon (:).
- 5 : No two case constants are identical.
- 6 : The case labels must be constants.

Example

```
#include<stdio.h>
main()
{
int a;
printf("Please enter a no between 1 and 5: ");
scanf("%d",&a);
switch(a)
{
case 1:
printf("You chose One");
break;
case 2:
printf("You chose Two");
break;
case 3:
printf("You chose Three");
break;
case 4:
printf("You chose Four");
break;
case 5: printf("You chose Five.");
break;
default :
printf("Invalid Choice. Enter a no between 1 and 5");
break;
}
}
```



