3. Logical NOT Operator

If the condition is true then the logical NOT operator will make it false and vice-versa. Below is the truth table for the logical NOT operator.

X	!X
0	1
1	0

Syntax:
!(condition_1 && condition_2)

```
#include <stdio.h>
 int main()
{
 int a = 10, b = 20;
   if !(a > 0 and b > 0)
  {
   printf("Both   values   are   greater   than
0\n");
  }
 else
  {
   printf("Both values are less than 0\n");
  }
  return 0;
 }
```

Conditional operator

→ The conditional operator in C is a kind of if-else stmt

→ it follows same algorithm as of if-else.

→ conditional operator takes less space than if-else and helps to write if-else in shortest way.

→ It is also known as Ternary operator.

Syntax:

```
                  ┌─ False ─┐
Variable = Exp1 ? Exp2 : Exp3;
               └─ True ┘
```

Visualized using if-else

```
if(Exp1)
{ variable = Exp2;
}
else
{ variable = Exp3;
}
```

Eg:

```
# include<stdio.h>
int main()
{ int n1=5, n2=10, max;
~~m>n ? printf("m is greater")~~.
: ~~printf("n is greater~~":
  max = (n1>n2)? n1 : n2;
 printf("Largest number is",
        n1, n2, max);
 return 0;
}
```

Logical Operators

Logical operators in C are used to combine multiple conditions/constraints. Logical Operators returns either 0 or 1, it depends on the expression result true or false. In C programming for decision-making, we use logical operators. We have 3 major logical operators in the C language:

- **Logical AND (&&)**
- **Logical OR (||)**
- **Logical NOT (!)**
- **Logical XOR(^)**

Types of Logical Operators

1. Logical AND Operator

If both operands are non zero then the condition becomes true. Otherwise, the result has a value of 0. The return type of the result is int. Below is the truth table for the logical AND operator.

X	Y	X&&Y
1	1	1
1	0	0
0	1	0
0	0	0

Syntax:

(condition_1 && condition_2)

```
#include <stdio.h>
int main()
{
 int a = 10, b = 20;
   if (a > 0 && b > 0)
  {
   printf("Both values are greater than 0\n");
  }
  else
  {
   printf("Both values are less than 0\n");
  }
  return 0;
}
```

2. Logical OR Operator

The condition becomes true if any one of them is non-zero. Otherwise, it returns false i.e, 0 as the value. Below is the truth table for the logical OR operator.

X	Y	X||Y
1	1	1
1	0	1
0	1	1
0	0	0

Syntax:
(condition_1 || condition_2)

```
#include <stdio.h>
int main()
{
 int a = 10, b = 20;
   if(a > 0 || b > 0)
  {
   printf("Both  values  are  greater  than
0\n");
  }
  else
  {
   printf("Both values are less than 0\n");
  }
  return 0;
}
```