Logical Operator

Sometimes single condition is not enough to determine if our criteria has been met.

Ex:
- If a person wants to login to Facebook, their username AND the password both need to match.

- For a person wants to watch a rated R movie, the person needs to be over 18 OR guardian needs to be present.

- A person is NOT on the authorized list of users, abort.
The AND Operator

- The ‘and’ operator in C++ is denoted with double ampersand sign -- &&
- Condition is true when both operands are true
- Condition is false when either operand is false

EX:
Detect for good age range: $0 < \text{age} \leq 100$
- positive and 100 and under

```cpp
if ( \text{age} > 0 ){
    if ( \text{age} \leq 100 ){
        // good age
    }
}
// using and operator
if ( \text{age} > 0 \: \&\& \: \text{age} \leq 100 ){
    // good age
}
```
/* ask user for a number
   print a message “It’s a 2 digit number!” if
   the input a 2 digit number
*/
#include <iostream>
using namespace std;
int main()
{
    int num;
    cout << “Please enter a number: “;
    cin >> num;

    if ( num >= 10 && num < 100 )
    {
        cout << “It’s a 2 digit number!”;
    }

    return 0;
}
**AND operator**

The condition evaluates to true ONLY when both operands are true.

<table>
<thead>
<tr>
<th>Operand 1</th>
<th>Operand 2</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
The OR Operator

- The ‘or’ operator in C++ is denoted with double pipe symbol -- ||
- Condition is false when both operands are false
- Condition is true when either operand is true

EX:
Detect for bad age range:
- negative or over 100
if ( age < 0 ){
    //bad age
}
if ( age > 100 ){
    //bad age
}
//using or operator
if ( age < 0 || age > 100 ){
    //bad age
}
program.cpp

/* ask user for a name
print a message "My bestie!" if the input
name is Tina or TINA
*/
#include <iostream>
using namespace std;
int main()
{
    string name;
    cout << "Please enter your name: ";
    cin >> name;

    if ( name == "Tina" || name == "TINA" ){
        cout << "My bestie!"
    }

    return 0;
}
OR operator

<table>
<thead>
<tr>
<th>Operand 1</th>
<th>Operand 2</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

The condition evaluates to true when Either operand is true.
The NOT Operator

- The ‘not’ operator in C++ is denoted with exclamation mark -- !
- Condition is false when operand is true
- Condition is true when operand is false

EX:
Detect for bad age range:
- not in range 0 < age <= 100

//using not operator
if ( !( age > 0 && age <= 100 ) ){
    //bad age
}

//bad age

program.cpp

/* ask user for a number
print a message “Oops, not this time. ” if
the input number is not ending with digit 7
*/
#include <iostream>
using namespace std;
int main()
{
    int num;
    cout << "Please enter a number: ";
    cin >> num;

    if ( !(num % 10 == 7))
    {
        cout << "Oops, not this time. “;
    }
}
return 0;
Not operator

<table>
<thead>
<tr>
<th>Operand</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

The condition negate the operand.