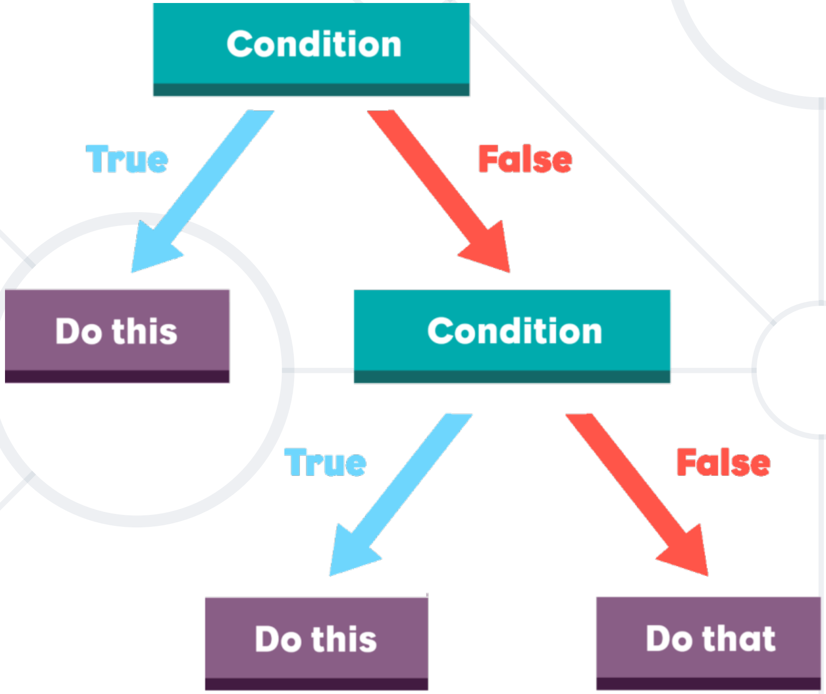
Complex Conditional Statements

Nested Conditional Statements, Logical Operators, Switch-case



Softuni Team

Technical Trainers

SoftUni

Table of Contents

SoftUni

1. Review of the Previous Lesson
2. Complex Conditional Statements: **Introduction**
3. Nested `if-else` Statements
4. Logical Operators: `&&`, `||`, `!`
5. Conditional Statement "`switch-case`"
   - Multiple Labels in Switch-Case

Have a Question?

SoftUni

sli.do

#prgm-for-qa

Review

Conditional Statements

Comparison Operators

- Comparison operators work for **numbers**
  - Equal to (==), not equal to (!=)
  - Greater than (>), greater than or equal to (>=)
  - Less than (<), less than or equal to (<=)

```
Console.WriteLine(5 != 5); // False
```

```
Console.WriteLine(5 < 6);  // True
```

```
Console.WriteLine(5 >= 5); //True
```

Conditional Statements

- The **if-else** statement can be in a series

```
if (…)
  // Some code
else if (…)
  // Another code
else
  // Other code
```

```
if ()
else if()
…
else
```

Nested Conditions

If-Else Inside Another If-Else

Nested Conditional Statements

- An `if-else` statement can be **nested** within another `if-else` statement

```
if (expression) {
  if (nested expression)
    // Some code
  else
    // Other code
}
```

Nested Conditional Statements (2)

- Only if the first condition is **true** the nested one is checked

```
if (expression) {
  if (nested expression)
    // Some code
  else
    // Other code
}
```

Executes when the nested expression is false

- **Deep nesting** is not recommended
  - Use up to **3 nested levels** for more readable code

Problem: Marketplace

- Read a **product** and **day** from the console
- Print the **price**, formatted to 2nd digit, based on the price table:

Product	Weekday	Weekend
Banana	2.50	2.70
Apple	1.30	1.60
Kiwi	2.20	3.00

```
Kiwi
Weekday
```

→

```
2.20
```

```
Banana
Weekend
```

→

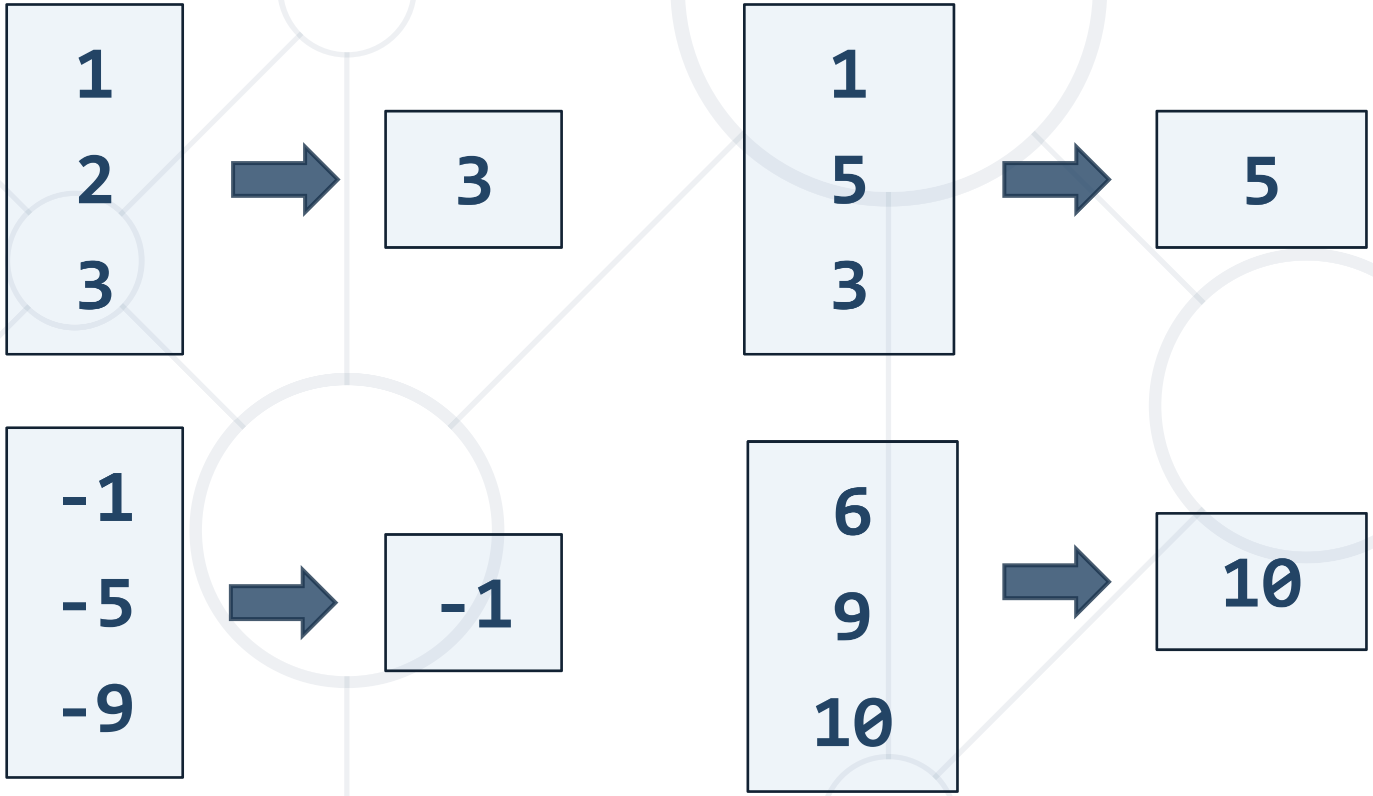
```
2.70
```

Solution: Marketplace

```
if (product == "Banana")
   if (dayOfWeek == "Weekday")
      Console.WriteLine("2.50");
   else
      Console.WriteLine("2.70");
else if (product == "Apple")
   if (dayOfWeek == "Weekday")
      Console.WriteLine("1.30");
   else
      Console.WriteLine("1.60");
// TODO: the same logic for "kiwi"
```

Problem: Largest Number out of Three

- Write a program, that reads **3 integer numbers** from the console and prints **the largest** of them



```
1
2
3
```

```
3
```

```
1
5
3
```

```
5
```

```
-1
-5
-9
```

```
-1
```

```
6
9
10
```

```
10
```

Solution: Largest Number out of Three

```
if (first > second)
  if (first > third)
    Console.WriteLine(first);
  else
    Console.WriteLine(third);
else
  if (second > third)
    Console.WriteLine(second);
  else
    Console.WriteLine(third);
```

first > second
first > third

third >= first > second

second >= first
second > third

third >= second >= first

NOT (!)
AND (&&)
OR (||)

Logical Operators

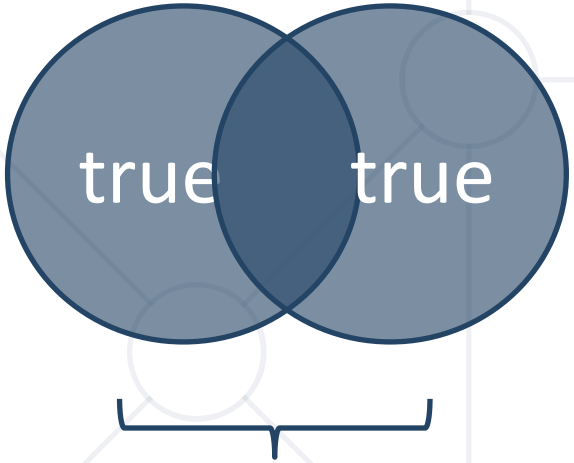
Checking Complex Conditions

Logical Operators

- Logical checks are based on **logical conditions**
- The **logical operators** in C# are:
  - Logical **AND** (**&&**)
  - Logical **OR** (**||**)
  - Logical **negation** (**!**)
- **Brackets ( )** change the order

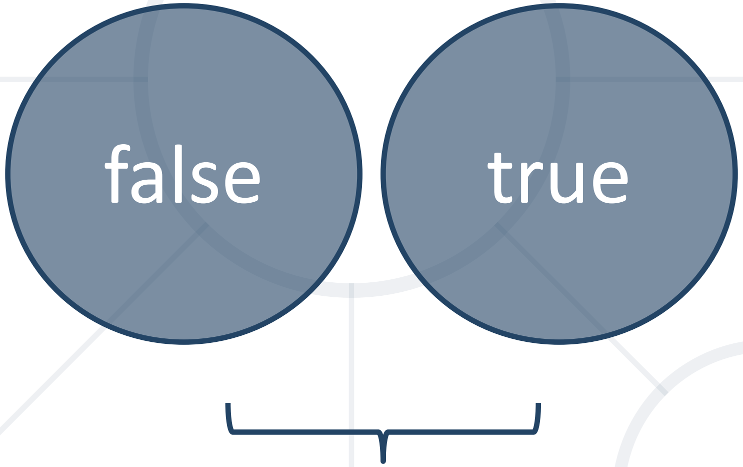
Logical Operators: Explanation

"&&" - AND



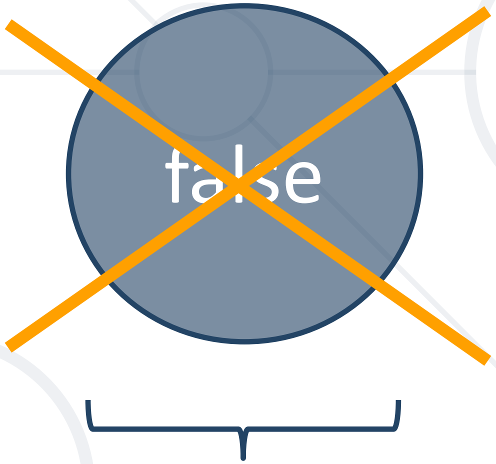
Both conditions must be true

"||" - OR



One condition must be true

"!" - NEGATION



Logical negation

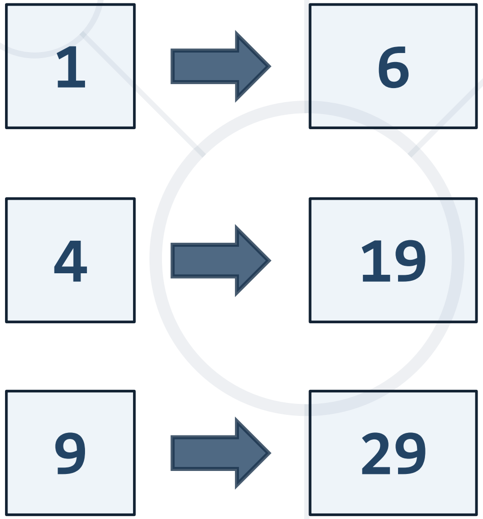
Logical AND (&&)

- Returns the Boolean value **true** if all of the operands are **true** and **false** otherwise
- Example: check **number** is in the following **range [100; 200]**

```
if (number >= 100 && number <= 200)
{
  Console.WriteLine("Number is in range");
}
```

Problem: Bonus Score

- Write a program to **add bonus** to given points
  - If points are between 0 and 3 (inclusive), adds 5 to the points
  - If points are between 4 and 6 (inclusive), adds 15 to the points
  - If points are between 7 and 9 (inclusive), adds 20 to the points



1 → 6

4 → 19

9 → 29

Solution: Bonus Points

```
int points = int.Parse(Console.ReadLine());
if (points >= 0 && points <= 3)
  points += 5;
else if (points >= 4 && points <= 6)
  points += 15;
else if (points >= 7 && points <= 9)
  points += 20;
Console.WriteLine(points);
```

Logical OR (||)

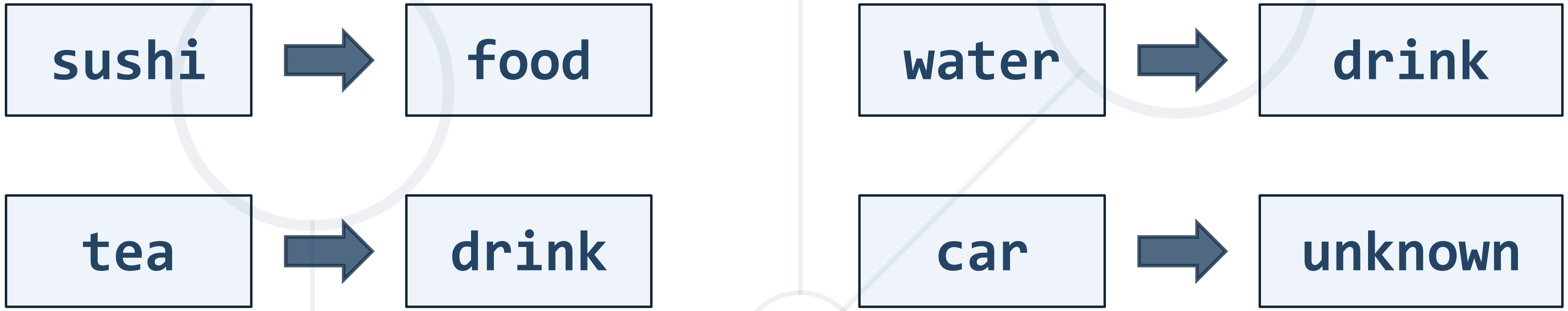
- The result of the expression is `true` if one of the operands is `true`, otherwise the result is `false`

```
product == "tea" || product == "water"
```

- Problem: **check for food or drink**
  - Read single line and print "`drink`", "`food`" or "`unknown`"
  - Foods: `curry`, `noodles`, `sushi`, `spaghetti`
  - Drinks: `tea`, `water`, `coffee`
  - Everything else is `unknown`

Problem: Food or Drink

- Problem: **check for food or drink**
  - Read single line and print "`drink`", "`food`" or "`unknown`"
  - Foods: `curry`, `noodles`, `sushi`, `spaghetti`, `bread`
  - Drinks: `tea`, `water`, `coffee`, `juice`
  - Everything else is `unknown`



sushi → food

water → drink

tea → drink

car → unknown

Solution: Food or Drink

```
string p = Console.ReadLine();
if (p == "curry" || p == "noodles" ||
    p == "sushi" || p == "spaghetti" || p == "bread")
  Console.WriteLine("food");
else if (p == "tea" || p == "water" || p == "coffee"
         || p == "juice")
  Console.WriteLine("drink");
else
  Console.WriteLine("unknown");
```

Logical NOT (!)

- Logical negation returns `true` when the operand is `false`, and `false` when the operand is `true`
- Example: **check for valid number**
  - A number is **valid** if is in the range [**100…200**] or is equal to **0**

```
bool isValid = (num >= 100 && num <= 200) || num == 0;
if (!isValid)
{
  Console.WriteLine("invalid");
}
```

Switch-Case

Checking Multiple Values for the Same Input

The Switch-Case Statement

- Used for choosing among a list of possibilities
- Alternative to an `if-else` statement

```
switch (selector) {
  case value1:
    statements;
    break;
  default:
    statements;
    break;
}
```

Switch-Case Example: Print Yes / No

```
string choice = Console.ReadLine();
switch (choice) {
  case "y":
     Console.WriteLine("Yes");
    break;
  case "n":
     Console.WriteLine("No");
    break;
  default:
     Console.WriteLine("Invalid response");
    break;
}
```

- Read a letter
  - "y" → print "Yes"
  - "n" → print "No"
  - Otherwise → print "Invalid response"

Multiple Labels in Switch-Case

Same Action for Several Values

Multiple Labels in Switch-Case

- Same logic may apply for more than one case

```
switch (selector) {
  case value1:
  case value2:
    statements;
    break;
  default:
    statements;
    break;
}
```

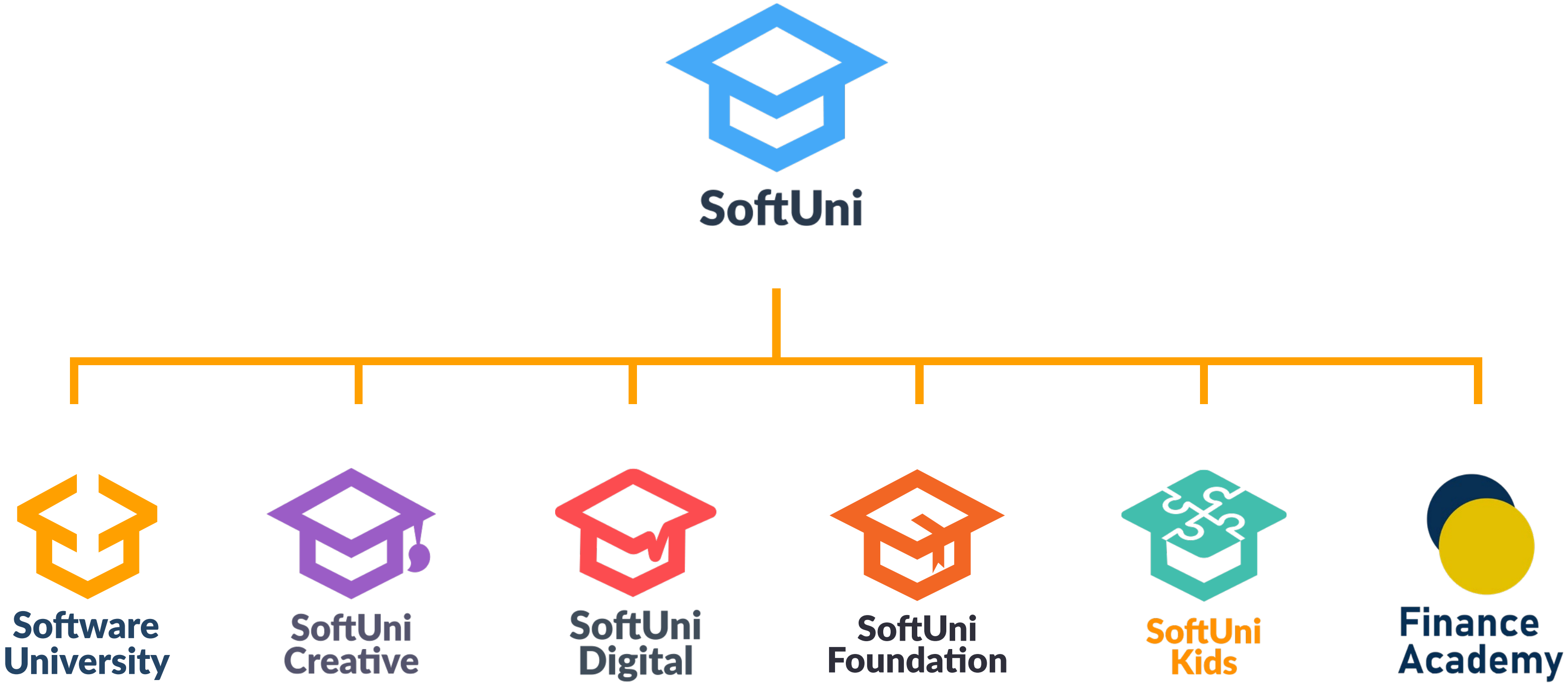
Multiple Labels: Example

```
string animal = Console.ReadLine();
switch (animal) {
  case "dog":
  case "cat":
     Console.WriteLine("mammal");
    break;
  default:
     Console.WriteLine("unknown");
    break;
}
```

Summary: Advanced Conditional Statements

- An `if-else` statement can be nested within another `if-else`
- **Logical operators** operate over boolean expressions
  - `&&` (and), `||` (or), `!` (not)
- The `switch-case` statement is an alternative to the `if-else`

Questions?


SoftUni
Software University
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids
Finance Academy

SoftUni Diamond Partners

SUPER
HOSTING
.BG

Coca-Cola HBC
Bulgaria

Flutter™
International

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT
KINGS

SOFTWARE
GROUP

BOSCH

Postbank
Решения за твоето утре

PHAR
VISION

SmartIT

DXC
TECHNOLOGY

createX

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License