03 Conditional statements

Programming fundamentals

YP0616 - YP0601

Elke Boonen & Tristan Vandevelde

Learning objectives (ECTS)

- **Basic principles** (types, operators, expressions) & **structures** (loop & if)
- **Arrays**, **lists**, **dictionaries**
- **Methods** and **functions**
- Basic principles of **OO**
- **Files**, in-and output **IO**
- **Exception** handling

Learning materials

- **Canvas** LMS https://thomasmore.instructure.com/
  - Presentations
  - E-Book: Fundamentals of Computer Programming with C#
  - Cheatsheet C#
  - Assigments (CodeGrade)
- **Online**
  - https://docs.microsoft.com/en-us/dotnet/csharp/
  - https://github.com/ElkeBoonen/ProgrammingFundamentals (code from slides)
  - https://github.com/ElkeBoonen/ProgrammingFundamentals-Students (code from class)
- **Software**
  - Visual Studio (Community) https://visualstudio.Microsoft.com/

Schedule

Before autumn break	**After autumn break**
01 Hello world	07 Exception handling
02 Variables & expression	08 Recap
03 If-structures	09 Collections
04 Loops	10 Methods
05 Files (IO)	11 OO
06 Arrays	12 OO
	13 Exam prep

Schedule is always subject to unexpected circumstances

Evaluation

- **1st term**
  - Permanent Evaluation (30 %):
    - CodeGrade exercises (each week, from week 02)
  - Computer Exam (70 %) use of cheatsheet only!
- **2nd term**
  - Computer Exam (100 %) use of cheatsheet only!

03 Conditional statements

- If…else
- To combine
- Switch
- Check by if

So many choices

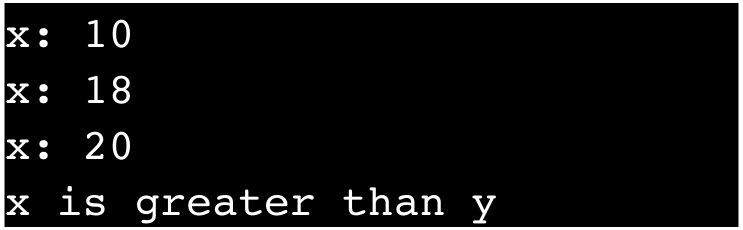
- An algorithm is a chronological sequence of statements
- We can **bypass chronology** by the use of conditions
- Take a **different statement depending on the result of a condition**
- Result condition always **true or false**

It should be logical

- C# inherits logical conditions from mathematics:
  - Less than: a < b
  - Less than or equal to: a <= b
  - Greater than: a > b
  - Greater than or equal to: a >= b
  - Equal to a == b
  - Not equal to: a != b

If I learn, I know

```
Console.Write("x:  ") ;
int x = Convert.ToInt32 (Console.ReadLine());
int y = 18;

if (x > y)
{
  Console.WriteLine("x is greater than y");
}
```



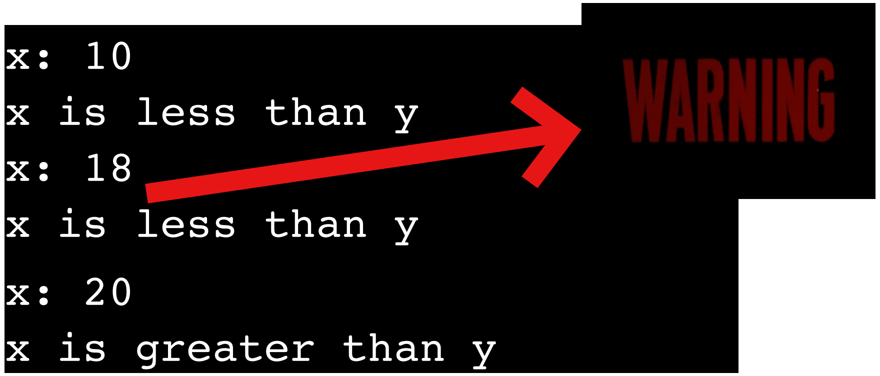
```
x:  10
x:  18
x:  20
x is greater than y
```

2 parts:

- condition (x > y)
- statement when true = show message 'x is greater than y'

If I learn, then I know, else I don't

```
Console.Write("x: ") ;
int x = Convert.ToInt32 (Console.ReadLine());
int y = 18;

if (x > y)
{
  Console.WriteLine("x is greater than y");
}
else
{
  Console.WriteLine("x is less than y");
}
```



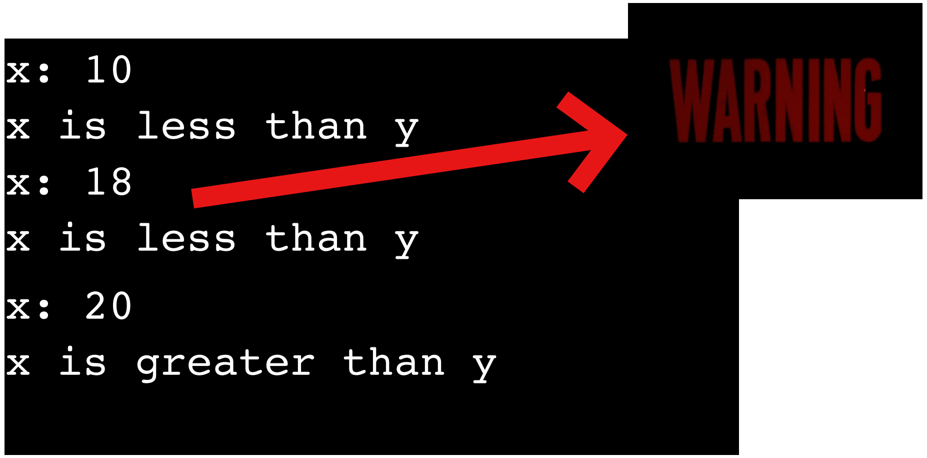
```
x: 10
x is less than y
x: 18
x is less than y
x: 20
x is greater than y
```

3 parts:

- condition (x > y)
- statement when true = show message 'x is greater than y'
- statement when false = show message ’x is less than y’

What if?

- What if two actions are not enough?
  - less, greater or equal to... 18?



```
x: 10
x is less than y
x: 18
x is less than y
x: 20
x is greater than y
```

- **More if-conditions to the rescue!**

Mentally somewhere else

- if, **else if**, else if, else if... else
- After another an else, another if can come!

```
Console.Write("x: ") ;
int x = Convert.ToInt32 (Console.ReadLine());
int y = 18;

if (x > y)
{
  Console.WriteLine("x is greater than y");
}
else if (x < y)
{
  Console.WriteLine("x is less than y");
}
else
{
  Console.WriteLine("x is equal to y");
}
```

```
x: 10
x is less than y
x: 18
x is equal to y
x: 20
x is greater than y
```

Lots of ways to code!

- We can also nest one if-structure in another
- There are many ways to reach the ’right’ code!

```
Console.Write("x:  ") ;
int x = Convert.ToInt32 (Console.ReadLine());
int y = 18;

if (x != y)
{
  if (x > y)
  {
        Console.WriteLine("x is greater than y");
  }
  else
  {
        Console.WriteLine ("x is less than y");
  }
}
else
{
  Console.WriteLine ("x is equal to y");
}
```

```
x:  10
x is less than y
x:  18
x is equal to y
x:  20
x is greater than y
```

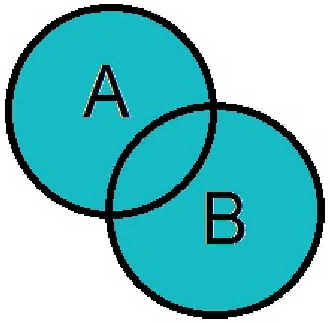
03 Conditional statements

- ~~If…else~~
- To combine
- Switch
- Check by if

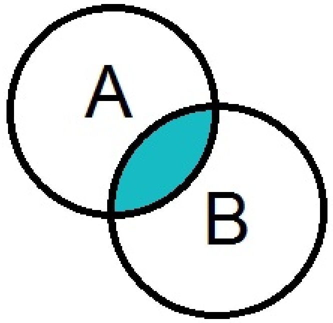
The art of combining

We can combine more than one condition

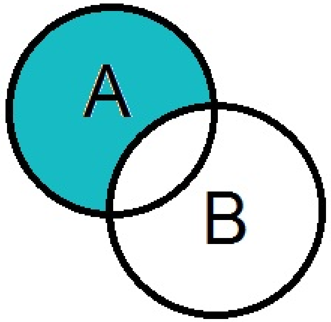
- **AND** &&: **both conditions** must be true
- **OR** ||: **at least one** condition must be true
- **NOT** !: **opposite** condition



A OR B



A AND B

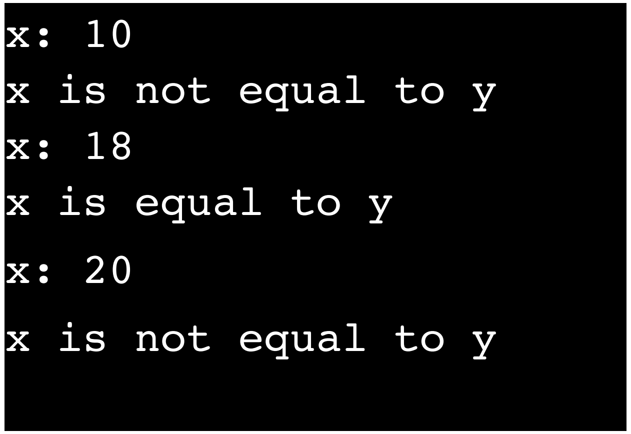


A NOT B

The art of combining

- Check both conditions by && (AND)
- Both conditions must me true!

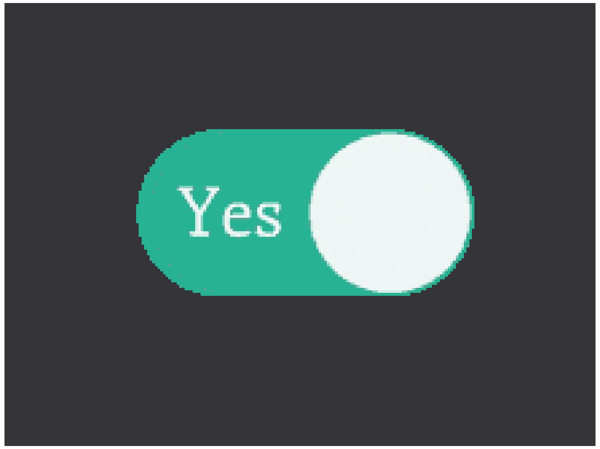
```
Console.Write("x: ") ;
int x = Convert.ToInt32 (Console.ReadLine());
int y = 18;

if ((x > y) || (x < y)) {
  Console.WriteLine ("x is not equal to y") ;
}
else
{
  Console.WriteLine ("x is equal to y") ;
}
```



```
x: 10
x is not equal to y
x: 18
x is equal to y
x: 20
x is not equal to y
```

True of false

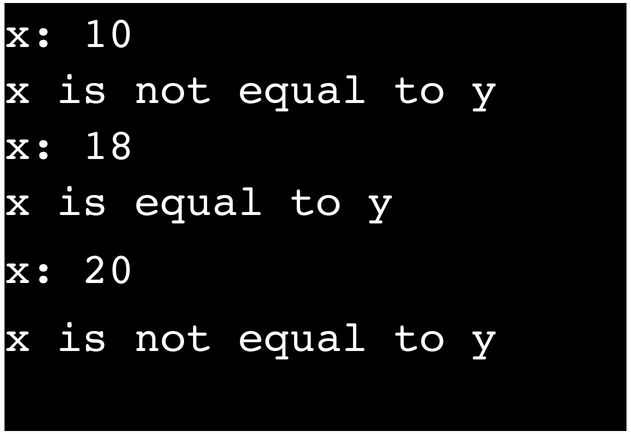
- An extra **value type: boolean (bool)**
- A bool has one of two possible values: true or false (1/0, yes/no)
- The result of a condition is always a bool
- A bool is very useful in state checking
  - Text filled in?
  - Timer started?
  - ...



If not true, then false

- Put the result of your condition in a bool-variable
- Check your condition with this variable (only true or false!)

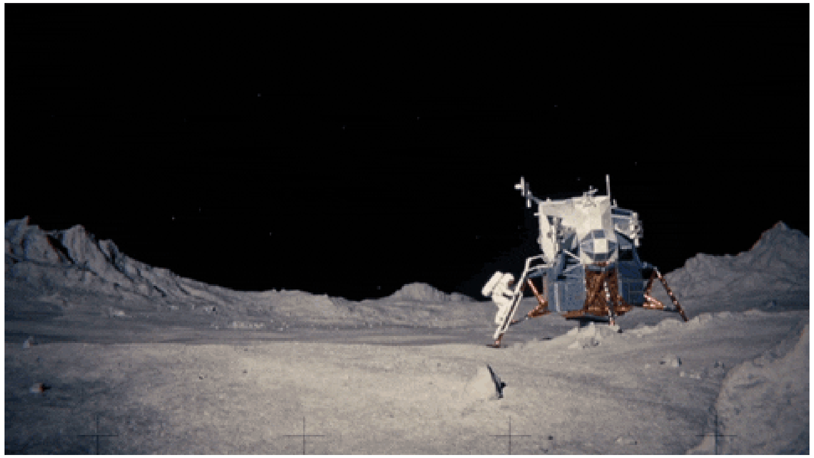
```
Console.Write("x:  ") ;
int x = Convert.ToInt32 (Console.ReadLine());
int y = 18;

bool result = x!=y ;

if (result)
{
  // or if ( result == true )
  Console.WriteLine("x is not equal to y") ;
}
else
{
  Console.WriteLine("x is equal to y") ;
}
```



```
x: 10
x is not equal to y
x: 18
x is equal to y
x: 20
x is not equal to y
```

Boolean flag

- Think of your boolean as a flag you can raise!
- Sometimes it is easier to simply set a boolean flag when a certain condition is detected, rather than have multiple nested if's!



03 Conditional statements

- ~~If…else~~
- ~~To combine~~
- Switch
- Check by if

If it is not a hit, switch

- What if **one condition has multiple possible results**?
  - eg: weekday can be 1 (Monday), 2 (Tuesday), 3...
- We can use several if-structures,

  but we could also use a **switch-structure!**

If vs switch

- Checking all days of the week with an if or switch
- Every **switch** can be **converted** to an **if-structure**, not the other way around!

```
DateTime today = DateTime.Now ;
int weekday = Convert.ToInt32 (today.DayOfWeek);

if (weekday == 1)
{
  Console.WriteLine ("It is Monday ") ;
}
else if (weekday == 2)
{
  Console . WriteLine ("It is Tuesday ") ;
}
else if (weekday == 3)
{
  Console.WriteLine ("It is Wednesday ") ;
}
// else if 4 - 5 - 6 - 7
else {
  Console.WriteLine("It is a crazy day !") ;
}
```

```
DateTime today = DateTime.Now ;
int weekday = Convert.ToInt32 (today.DayOfWeek) ;

switch (weekday)
{
  case 1: Console.WriteLine("It is Monday ") ;
        break; // when it is the case , break out!
  case 2: Console.WriteLine("It is Tuesday ") ;
        break;
  case 3: Console.WriteLine("It is Wednesday ") ;
        break;
  //cases 4 - 5 - 6 - 7
  default: Console.WriteLine("It is a crazy day !") ;
        break;
}
```

03 Conditional statements

- ~~If…else~~
- ~~To combine~~
- ~~Switch~~
- Check by if

Input must be idiot proof

- We can use **if-structures to check our input**
  - Is our input valid?
  - Is there input?
  - ...
- **Avoid the error message**

Not everything gets converted

```
Console.Write ("x: ") ;
int x = Convert.ToInt32(Console.ReadLine());
int y = 18;

if (x > y) {
  Console.WriteLine("x is greater than y");
}
```

```
x: twenty
```

System.FormatException: ’Input string was not in a correct format.’ because ’twenty’ is not an integer, so Convert.ToInt32 cannot work properly to convert the string to an integer!

Avoid the error message with an if!

```
Console.WriteLine("x: ") ;
string answer = Console.ReadLine();

int x; // parse answer , if it works -> x is ready
bool succes = Int32.TryParse (answer,out x);

int y = 18;

if (succes)
{
  if (x > y)
  {
        Console.WriteLine("x is greater than y");
  }
}
else {
  Console.WriteLine("Oops , crazy input !");
}
```

```
x: twenty
Oops, crazy input!
```

Practice makes perfect!

- Do your exercises, spend the hours!
- The better the exercises, the better the exam!

Say what? How many hours?

6 SP = 6 * 28 hours = 168 hours

Lessons = 12 * 3 hours = 36 hours

Exam = 2 hours

Exercise = 168-36-2 = 130 hours

Tomorrow land

- Big problem!
  Tomorrowland is throwing a party but the bouncer is sick, so they want to make a robot bouncer. This robot bouncer will scan person and let them know if they can enter the building based on there age/gender
- Phase 1:
  All persons who are older or equal than 18 and younger than 30 can enter

Age: 20, Gender: M	**can enter**
Age 17, Gender: F	**Can not enter**
Age 29, Gender: F	**Can enter**

Tomorrow land

- Oh no, there are way to much males at the party, the bouncer should be more strict who can enter
- Phase 2:

  All females who are older or equal than 18 and younger than 30 can enter

Age: 20, Gender: M	**can not enter**
Age 17, Gender: F	**Can not enter**
Age 29, Gender: F	**Can enter**

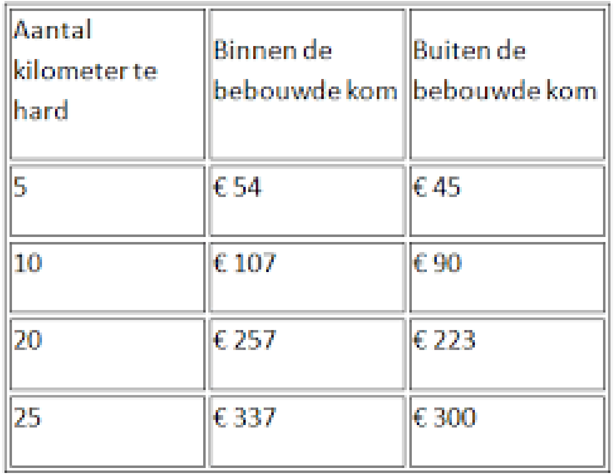
Tomorrow land

- Oh damn.
  The robot is made in by the maffia.
  It will still let women between 18 and 30 in, but if the guest gives 100EUR it will let them in.
- Phase 3:
  All females who are older or equal than 18 and younger than 30 can enter
  OR the bribe the robot by 100EUR

Age: 20, Gender: M Bribe-money: 100EUR	**can enter**
Age 17, Gender: F	**Can not enter**
Age 29, Gender: F	**Can enter**

The sound of the police

- The policer officer called and the speed camera is not working anymore. Lets help the police!
  Our code should register the speed and depending on the zone it should give a fine



Aantal kilometer te hard	Binnen de bebouwde kom	Buiten de bebouwde kom
5	€ 54	€ 45
10	€ 107	€ 90
20	€ 257	€ 223
25	€ 337	€ 300