

# Programming - C#

Finn Vilsbæk

[fvs@medieskolerne.dk](mailto:fvs@medieskolerne.dk)

# Emner for i dag

- - Typer (Types)
  - - Variabler (Variables)
  - - Konstanter (Constants)
  - - Tekststreng (Strings)
- 
- **Læs: *Learning C#, chapter 3***
  - **Chapter 3 excerpt: <http://msdn.microsoft.com/en-us/library/orm-9780596521066-01-03.aspx>**

# Typer – to slags:

- Intrinsic types: indbygget i sproget C#
  - Custom types: de typer, du selv skaber
- 
- Intrinsic types kan være numeric eller non-numeric
  - int, double, float, char, bool er alle intrinsic types
  - Objekter der har en intrinsic type kaldes *variables*
  - IAnimal, mammal, dog - entiteter, som vi skaber
  - C# er et **strongly typed (typestærkt)** sprog . Det betyder..?
  - Hvert objekt i et C# program skal have en specifik **type**

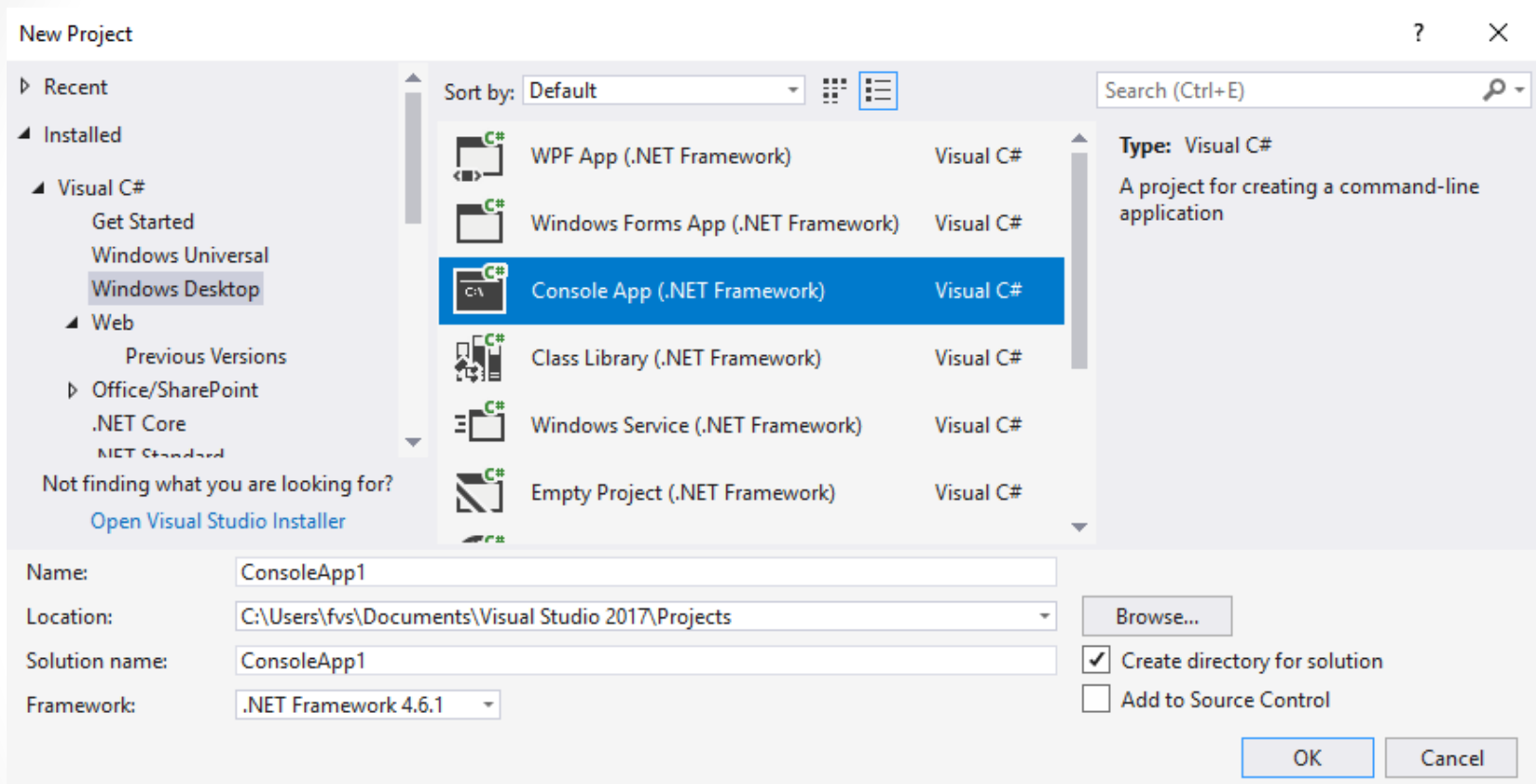
**Table 3.1. The intrinsic types built into C#**

C# type	Size (in bytes)	.NET type	Description
byte	1	Byte	Unsigned (values between 0 and 255).
char	2	Char	Unicode characters (a modern way of storing most characters, including international language characters).
bool	1	Boolean	True or false.
sbyte	1	SByte	Signed (values between -128 and 127).
short	2	Int16	Signed (short) (values between -32,768 and 32,767).
ushort	2	UInt16	Unsigned (short) (values between 0 and 65,535).
int	4	Int32	Signed integer values between -2,147,483,648 and 2,147,483,647.
uint	4	UInt32	Unsigned integer values between 0 and 4,294,967,295.
float	4	Single	Floating-point number. Holds the values from approximately $\pm 1.5 \times 10^{-45}$ to approximately $\pm 3.4 \times 10^{38}$ with seven significant figures.
double	8	Double	Double-precision floating-point. Holds the values from approximately $\pm 5.0 \times 10^{-324}$ to approximately $\pm 1.8 \times 10^{308}$ with 15 to 16 significant figures.
decimal	12	Decimal	Fixed-precision up to 28 digits and the position of the decimal point. This type is typically used in financial calculations. Requires the suffix "m" or "M" when you declare a constant.
long	8	Int64	Signed integers ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
ulong	8	UInt64	Unsigned integers ranging from 0 to approximately $1.85 \times 10^{19}$ .

# Intrinsic Types

- Numeric types – int, uint, float, double
  - Unsigned værdier (values) kan kun holde ‘positive numbers’
  - Signed værdier (values) kan holde både positive og negative numbers
- Non-numeric types – char and bool
  - char type:
    - kan repræsentere en simpel character: ‘A’
    - kan repræsentere en unicode character: ‘\u0041’
    - kan repræsentere en escape sequence: ‘\n’
  - bool type:
    - kan repræsentere true eller false, bruges til at sammenligne values

# Make a console app



# Øvelse 1

Brug `System.Console.WriteLine` til output af lønninger

- Skriv denne kode ind i main metoden, på een linie:

```
System.Console.WriteLine  
("Employee wages:\n First employee:\t{0}\n  
Second employee:\t{1}", myInt, myOtherInt);
```

Sæt den første lønning til 20000, og den anden lønning til 26000.

Tip: Tryk `Ctrl-F5` for at beholde konsollens output synligt.

# Variabler

- Definition: en *placeholder* for en *value* / en værdi.

```
int numOfClaws = 4;
```

- Teknisk set:
  - En instans af en intrinsic type, gemt i en 'named storage location'
- Man kan også bruge keywordet *var*, og lade C# *inferere* typen – men det er kun anbefalet hvor det er nødvendigt (implicite typer vil gøre din kode mindre læsbar)

```
var numOfClaws = 4;
```



# Variabler

- En speciel container for variabler: **struct**
- En struct er en datastruktur, som regel sammensat af flere forskellige stykker relaterede data, måske af forskellig type.
- “Declare once - reuse many times”.

```
struct route
{
    public orientation    direction;
    public double        distance;
}
```

# Konstanter: Constants

- Tre slags: *literals*, *symbolic constants*, og *enumerations*.
  - En literal constant er en værdi, ie. 64.
  - En symbolic constant kobler et navn til en konstant værdi.  
`const int FreezingPoint = 32;`
  - En enumeration er et sæt af logisk relaterede konstanter.  
Default type: int

```
enum DamageLevel
{
    Healthy = 95,
    Challenged = 50,
    Hurting = 25,
    Critical = 10,
    Wasted = 0,
}
```

# Øvelse 2

Mål: lær at bruge en enum

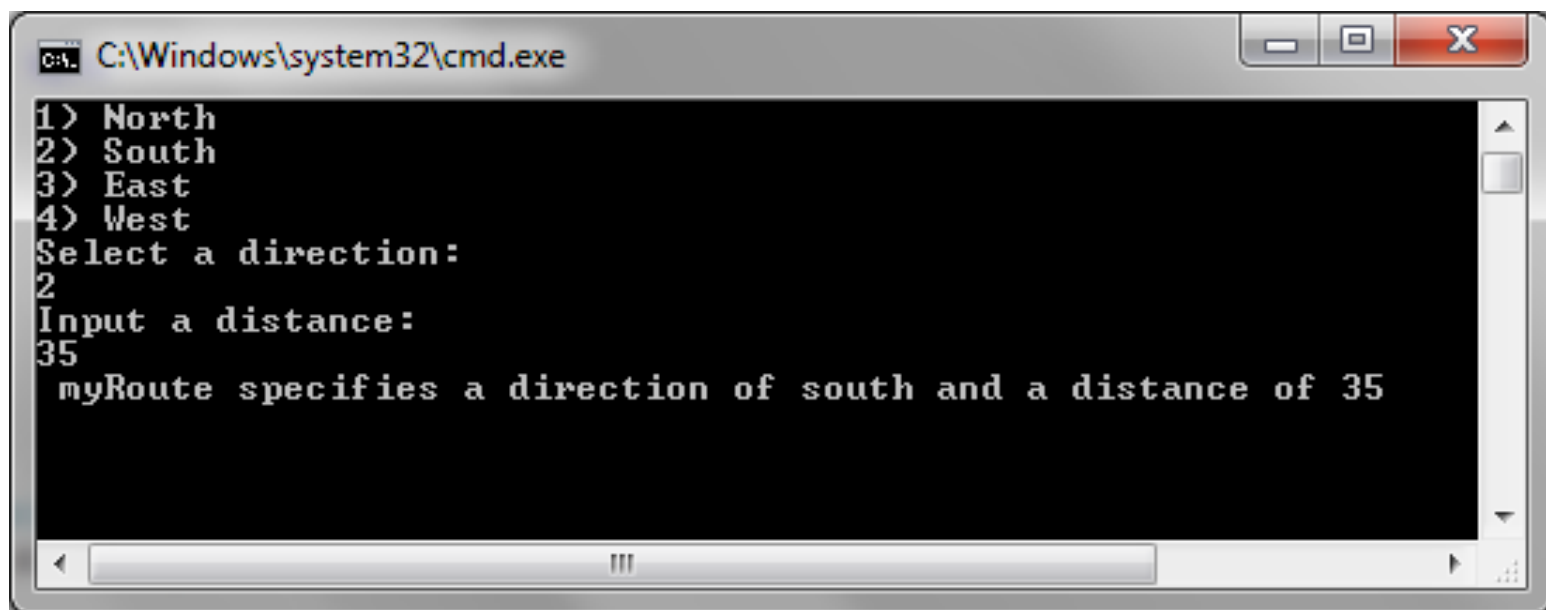
- Lav en ny console app, eller genbrug den fra øvelse 1.
- Lav en constant enumeration 'Months' der holder konstanter for hver måned i årets løb.
- Giv hver måned den værdi, der svarer til dens numeriske plads i kalenderen, så Januar bliver 1, Februar bliver 2 etc.
- Output værdien for Juni, med beskeden:  
    'Værdien for Juni er: '

# Øvelse 3

- Lav en ny konsol application.
- Skriv hand-out koden ind i din konsol-app, og prøv forskellige inputs:
- Input et nummer imellem 1 og 4 for at vælge en orientation
- Input et andet nummer for at specificere en distance

# Øvelse 3

Konsollen burde nu vise dig følgende output:

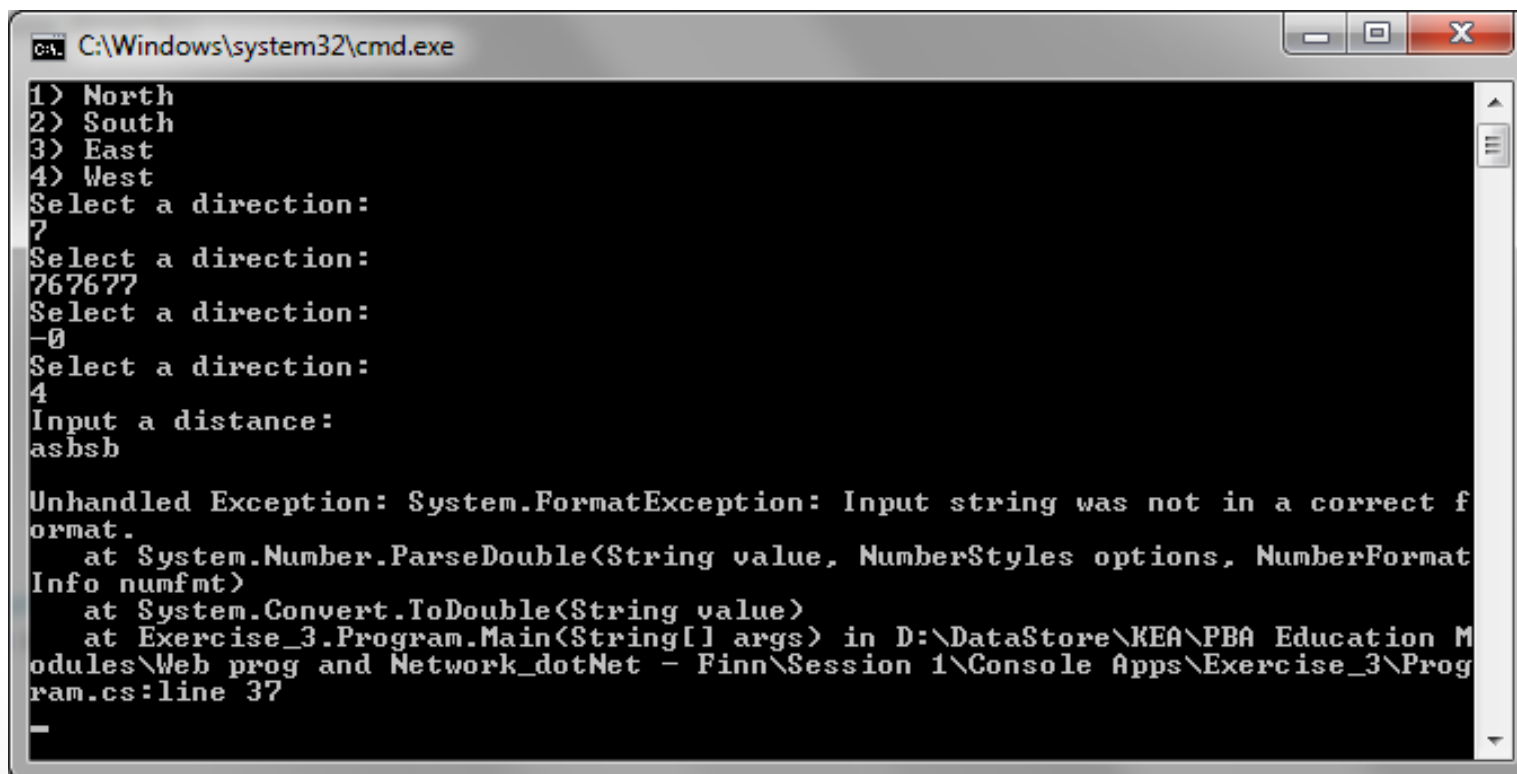


```
C:\Windows\system32\cmd.exe
1) North
2) South
3) East
4) West
Select a direction:
2
Input a distance:
35
myRoute specifies a direction of south and a distance of 35
```

- Hvad sker der, hvis input er 'forkert' – ie. Ikke et tal?

# Øvelse 3

- Input 'asbsb':



```
C:\Windows\system32\cmd.exe
1> North
2> South
3> East
4> West
Select a direction:
7
Select a direction:
767677
Select a direction:
-0
Select a direction:
4
Input a distance:
asbsb

Unhandled Exception: System.FormatException: Input string was not in a correct format.
   at System.Number.ParseDouble(String value, NumberStyles options, NumberFormatInfo numfmt)
   at System.Convert.ToDouble(String value)
   at Exercise_3.Program.Main(String[] args) in D:\DataStore\KEA\PBA Education Modules\Web prog and Network_dotNet - Finn\Session 1\Console Apps\Exercise_3\Program.cs:line 37
```

- Kan du tænke dig til en måde at håndtere sådanne input fejl på?

# Strings

- Syntaks test i VS 2017

Vil den følgende kode compile? Hvorfor, eller hvorfor ikke?

```
string[] blah = new string[5]  
string[5] = 5th string
```

# Strings

```
// Put a sentence into an array of separate words
string myString = "Luke, don't go so close to that power generator..";
char[] separator = { ' ' };
string[] myWords;
myWords = myString.Split(separator);
foreach (string word in myWords)
{
    Console.WriteLine("{0}", word);
}
```



# Strings

Øvelse 4: 'The yes man':

Skriv en console app som:

- 1) Acceptorer en string som input fra brugeren
- 2) Erstatte alle forekomster af 'no' med 'yes!'
- 3) Outputter den resulterende tekststreng til konsollen

# Strings

Hints til the yes man :

- Brug `Console.ReadLine()`; for at få fat i brugerens input
- Separer ordene i inputtet ved at kigge efter spaces ( ' ')
- Brug et foreach loop til at iterere igennem brugerens input

# Strings

## Øvelse 5: Hamlet i bakgear

Skriv et program, der kan vende rækkefølgen af ordene i nedenstående tekst om, og output den omvendte string til konsollen.

- *“To be, or not to be: That is the question: Whether ’tis nobler in the mind to suffer the slings and arrows of outrageous fortune, or to take arms against a sea of troubles, and by opposing end them?”*

# Strings

## Hints til Hamlet: 1

- Tricket til at kunne vende rækkefølgen af ord om i en givet sætning er at splitte strengen ad i dens 'component words', og derefter skal ordene samles igen til en ny streng.
- At splittte en string er nemt; det er hvad **Split()** metoden er bygget til. I dette tilfælde ved vi, at vores string indeholder spaces (mellemrum), kommaer, og kolon-tegn, så vi er nødt til at bruge alle tre som *delimiters*.

# Strings

Hints til Hamlet - 2:

- **Split()** returnerer et array, hvis klasse heldigvis har en statisk metode **Reverse()**, så man kan rimeligt nemt få ordene i omvendt rækkefølge ved at kalde denne metode som følger: `Array.Reverse(myStringArray)`.
- Når du har dit omvendte array, kan du bruge et **StringBuilder** objekt til at skabe den nye, omvendte string med.
- For at lægge en space (et mellemrumstegn) efter hvert enkelt ord, kan du med fordel sætte et foreach loop op, som bruger `StringBuilder` klassens **AppendFormat** metode.

# Strings

## Øvelse 6: Fly me to the moon

Skriv et program, der tæller op hvor mange gange ordet 'the' forekommer i nendestående tekst. Skriv derefter antallet af 'the' forekomster ud til konsollen.

*"We choose to go to the moon. We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard, because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one which we intend to win, and the others, too."*