



UPPSALA
UNIVERSITET



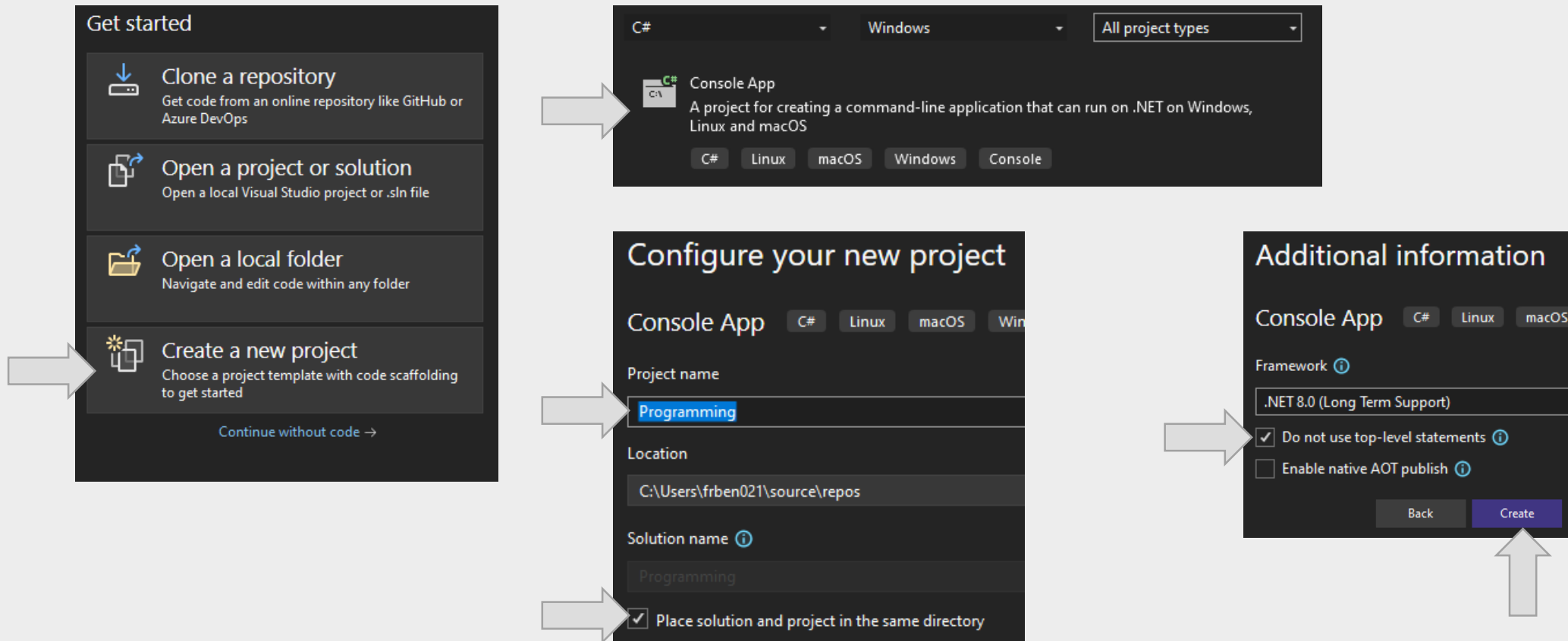
UPPSALA
UNIVERSITET

Introduktion till Visual Studio

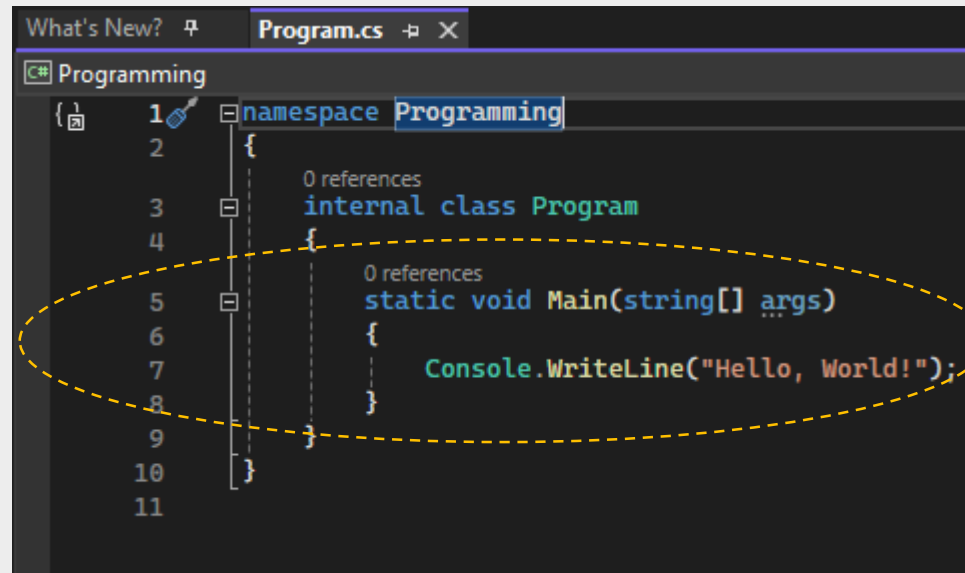
Tilldelning, Utskrift, Konvertering och Selektion

Visual Studio

- Hur man startar ett projekt av rätt typ?



Main metoden



The screenshot shows a C# code editor window titled "Program.cs". The code is as follows:

```
1 namespace Programming
2 {
3     0 references
4     internal class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Hello, World!");
10        }
11    }
```

A dashed yellow oval highlights the `static void Main` method and its body, which is the entry point of the program.



Tilldelning

- Tilldelning används för att lagra ett värde i en variabel.

```
1 namespace Programming
2 {
3     0 references
4     internal class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             string hello = "Hello!"; // textsträng
10            bool x = true; // boolean
11            int one = 1; // integer
12            int sum = 1 + 2; // integers, sum = 3
13            double dec = 1.4; // decimaltal
14            char c = 'a'; // tecken
15
16            var result = "Var String " + 10; // result = "Var String 10"
17        }
18    }
19 }
```

- Om **var** används så bestämmer det till höger om likamedtecknet (=) vilken typ variabeln får.
- Bör användas sparsamt då det finns ett värde i att hålla koll på variabelns typ.



Tilldelning ReadLine

- För att läsa input från användaren kan man använda `Console.ReadLine()`
- Variabeln input kommer att innehålla allt som användaren skriver fram till dess att denne trycker på enter.

```
1 namespace Programming
2 {
3     0 references
4     internal class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             string input = Console.ReadLine();
10        }
11    }
```

`string? Console.ReadLine()`
Reads the next line of characters from the standard input stream.

Returns:
The next line of characters from the input stream, or `null` if no more lines are available.

Exceptions:
`IOException`
`OutOfMemoryException`
`ArgumentOutOfRangeException`

[GitHub Examples and Documentation \(Alt+O\)](#)

CS8600: Converting null literal or possible null value to non-nullable type.

[Show potential fixes \(Alt+Enter or Ctrl+.\)](#)



Utskrift i konsolen

- För att istället skriva ut information till konsolen så kan man använda `Console.WriteLine()`
`Console.WriteLine("Ange ett heltal: ");`
- Man kan även skriva ut variabler i konsolen, oavsett dess datatyp. T.ex. enligt:
`int ten = 10; Console.WriteLine(ten);`
- WriteLine nyttjas för att skriva ut en rad i konsolen medan Write istället kan nyttjas för att skriva ut flera saker på samma rad.

```
Console.WriteLine("Ange ett heltal: ");
```

```
Console.Write("Ange ");
```

```
Console.Write("ett heltal: ");
```



Utskrift i konsolen

```
1 namespace Programming
2 {
3     0 references
4     internal class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             int ten = 10;
10            Console.WriteLine(ten);
11        }
12    }
13 }
```

▶ Programming ▾

Microsoft Visual Studio Debug Console

```
10
C:\Users\frben021\source\repos\Programming\bin\Debug\net8.0\Programming.exe (process 20512) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```



Konvertering av input

- Allt som tas emot av `Console.ReadLine()` är av datatypen string.
- Vi kan ändra datatypen string genom att konvertera till den datatyp vi behöver.

```
1 namespace Programming
2 {
3     0 references
4     internal class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             string input1 = Console.ReadLine();
10            string input2 = Console.ReadLine();
11            int intOne = Convert.ToInt32(input1);
12            int.TryParse(input2, out int intTwo);
13
14            Console.Write("Input 1 + Input 2 = ");
15            Console.WriteLine(intOne + intTwo);
16        }
17    }
18 }
19 }
```

```
Microsoft Visual Studio Debug Console
2
3
Input 1 + Input 2 = 5

C:\Users\frben021\source\repos\Programming
To automatically close the console when de
le when debugging stops.
Press any key to close this window . . .
```



Selektion: if

- Selektioner evaluerar ett booleskt uttryck
- Ett booleskt uttryck resulterar i sant (true) eller falskt (false)
- De selektionsfunktioner som vi använder oss utav är:

```
0 references
static void Main(string[] args)
{
    if(true)
    {
        // Om sant, gör något
    }
    else if (false)
    {
        // Om falskt, gör något annat
    }
    else
    {
        // Annars gör vi det här
    }
}
```

```
0 references
static void Main(string[] args)
{
    if(true)
    {
        // Om sant, gör något
    }
    else if (false)
    {
        // Om falskt, gör något annat
    }
    else
    {
        // Annars gör vi det här
    }
}
```

CS0162: Unreachable code detected
IDE0035: Unreachable code detected

Learn / .NET / C# /

Compiler Warning (level 2) CS0162

Article • 09/15/2021 • 8 contributors

Unreachable code detected

The compiler detected code that will never be executed.

Example

The following example generates CS0162:

```
C#
// CS0162.cs
// compile with: /W:2
public class Program
{
}
```

Learn / .NET /

Ta bort kod som inte kan nås (IDE0035)

Artikel • 2024-05-08 • 2 deltagare

I den här artikeln

[Översikt](#)
[Alternativ](#)
[Exempel](#)
[Ignorera en varning](#)
[Se även](#)

Property	Värde
Regel-ID	IDE0035
Title	Ta bort kod som inte kan nås
Kategori	Format



GÄVLE
UNIVERSITET

Selektion: if

- Som tidigare sades så evaluerar selektioner booleska uttryck.
- If-satsen är något som används väldigt ofta.

Inom parentesen evalueras uttrycket

```
0 references
static void Main(string[] args)
{
    if(1 + 3 == 4)
    {
        Console.WriteLine("1 + 3 = 4");
    }

    if(1 + 3 != 5)
    {
        Console.WriteLine("1 + 3 är inte lika med 5");
    }
}
```

Notera att man använder == för att kontrollera om något är lika

Notera att man använder != för att kontrollera om något **inte** är lika



Selektion: if

- Det finns fler booleska uttryck som man kan använda i en if-sats.

```
0 references
static void Main(string[] args)
{
    bool t = true;
    bool f = false;

    if(t && f)
    {
        Console.WriteLine("Både t och f är sanna.");
    }

    if(t || f)
    {
        Console.WriteLine("t eller f är sann.");
    }
}
```

t och f tilldelas true respektive false

Här evalueras t och f med && som betyder och/and, && evaluerar till sant när både t och f är sanna.

Här evalueras t och f med || som betyder eller/or, || evaluerar till sant när minst en av t och f är sann.



Selektion: if, else if, else

- Om man sätter flera if-satser efter varandra så kommer alla att evalueras vilket kan resultera i logiska fel.
- En else if-sats används för att endast evaluera tills ett fall resulterar i true eller alla resulterar i false.
- Om alla resulterar i false kan man fånga upp detta med en else-sats.

```
0 references
static void Main(string[] args)
{
    bool t = true;
    bool f = false;

    if(t && f)
    {
        Console.WriteLine("Både t och f är sanna.");
    }
    else if(t || f)
    {
        Console.WriteLine("t eller f är sann.");
    }
    else
    {
        Console.WriteLine("Inga av de tidigare avaluerades till true");
    }
}
```

Är både t och f sanna?

Om den tidigare evalueringen resulterade i false evaluerar vi denna.

Om de tidigare evalueringarna resulterade i false körs den efterföljande koden.



Uppgift 1

- Skriv ett program som läser in ett namn från användaren genom konsolen. Om användaren inte skriver in något så skall "Hello World" skrivas ut i konsolen, annars ska "Hello" följt av användarens namn skrivas ut.



Uppgift 2

- Skriv ett program som tar emot två inputs och kontrollerar om dessa två, tillsammans, utgör summan 5. Om så är fallet så ska "Yes" skrivas ut i konsolen, annars "Nope".



Uppgift 3

- Skriv ett program som tar emot två heltal och kontrollerar ifall dessa har samma värde. Om de har samma värde så skall texten "The integers are equal" skrivas ut. Om heltalen inte är lika skall summan av heltalen skrivas ut. Men om något utav heltalen har värdet 0 så skall texten "At least one integer is zero" skrivas ut.



Slut



UPPSALA
UNIVERSITET