



Campus Nyköping

PROGRAMMERING I .NET C# 1 - 02

Grunderna

Förra gången

- Sprint review 2
- Inlämning
 - Kanban-tavla/Trello-board
 - Sprint retrospect
- Eventuell VG-uppgift

Idag

- Ny kurs!
- Grunderna
 - Programmering
 - .NET
 - C#

Vad är ett program?

- Program är att man **ger instruktioner till en dator** som sedan berättar steg för steg hur den utför en uppgift som programmeraren har gett.
- Hur man gör för att datorn ska förstå vad programmeraren vill göra för något, då tar man hjälp via ett **programmeringsspråk** och skapar därefter ett program.

```
10 print "Hej världen"  
20 goto 10
```



Programmeringens faser

- Definiera ett problem =Specifikation
- Planera lösningen =Design
 - Agil utveckling
 - Hitta lämplig metod
 - Hitta lämpliga datastrukturer
- **Skriv kod** =Implementation
- **Fixa fel i koden** =Test och Debug
- Gör kunden glad =Driftsättning

Vad behöver du för att programmera?

- Ett problem att lösa
- Kunskap om ett programmeringsspråk
 - C#
- Utvecklingsmiljö, kompilatorer, SDK
 - Visual Studio, .NET SDK (Software development Kit)
- En uppsättning med standardklasser
 - Microsoft .NET FCL (framework class library)
- Hjälp, documentation
 - Microsoft Docs - [C# docs - get started, tutorials, reference.](#) | Microsoft Docs

Vad är C#?

- Programmeringsspråk
 - En syntax varmed vi kan ge instruktioner till en dator
- Egenskaper hos C#:
 - Modernt
 - Extremt kraftfullt
 - Lätt att lära
 - Lätt att läsa och förstå
 - Objektorienterat

Vad finns i ett program?



Indata

- Någon form av data som matas in av användaren.

Vad heter du?

Anders

- Indata från databaser, sensorer, knapptryckningar

Utdata

- Programmet kommer fram till ett resultat, som sedan skrivs ut på skärmen.

```
Console.WriteLine("Hejsan svejsan");
```

Variabler

- En variabel är ett sätt att lagra ett värde.
- Det kan vara ett tal, en text-sträng eller mycket mer komplicerade datatyper.

```
int x = 23;
```

```
string minFinaText = "Hejhopp";
```

```
double temp = 10.2;
```

Eller...

```
var variable = "Hejsan";
```

Loopar

- De flesta program gör samma sak flera gånger, repeterar.

```
For (i=1; i < 10; i++)  
{  
    Console.WriteLine(i)  
}
```

Villkor

- Anger att visa saker i programmet bara ska utföras om ett specifikt villkor uppfylls.

```
For (x = 1; x < 10; x++)  
{  
    if (x < 5)  
    {  
        Console.WriteLine (x) ;  
    }  
}
```

Operatorer

- Värdet i en variabel kan ändras enligt matematiska "formler".

```
int a, b, c;
```

```
a = 7;
```

```
b = a;
```

```
c = b + 3;
```

```
b = a + b * c;
```

Hello world I C#

```
using System;
```

Inkluderar
standard
namespace
"System"

Skapar vårt eget
namespace
"ConsoleApp1"

```
namespace ConsoleApp1
```

```
{
```

```
class Program
```

```
{
```

```
static void Main(string[] args)
```

```
{
```

```
    Console.WriteLine("Hejsan världen");
```

```
}
```

```
}
```

```
}
```

Skapar en klass
som heter
Program

Definierar en
metod som heter
"Main()".
Det är
programmets
startpunkt

Skriv text på
skärmen genom att
anropa metoden
"WriteLine" i klassen
"Console", som finns
i namespace
"System"

Dålig formattering av koden

- Kod ska kunna läsas av människor, inte bara av datorer

```
using
System

                                ;

    class      HelloCSharp      {

        static

void      Main(                ) { Console . WriteLine ("Hello,
C#"      ) ;Console.

    WriteLine      (                "Hello again"

                                ;}}
```

Sådan formatering
gör koden omöjlig
att läsa

C#-kod formattering

The diagram illustrates C# code formatting rules using a code snippet and callout boxes:

```
using System;  
class HelloCSharp  
{  
    static void Main()  
    {  
        Console.WriteLine("Hello, C#");  
    }  
}
```

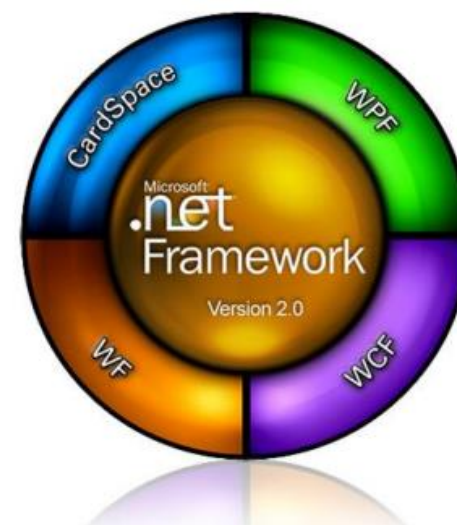
Callout boxes provide the following explanations:

- Klassnamn har PascalCase med Stor första bokstav** (Class name has PascalCase with capital first letter) - points to `class HelloCSharp`
- Tecknet { står ensamt på en ny rad** (The brace { stands alone on a new line) - points to the opening brace of the class
- Blocket efter { är indraget med en TAB** (The block after { is indented with a TAB) - points to the `static void Main()` block
- Avslutande } står under motsvarande {** (Closing brace } stands under the corresponding opening brace {) - points to the closing brace of the class



Vad är .NET?

Hette tidigare .NET Framework



Vad är .NET?

- En miljö för att köra .NET-program
- Ett kraftfullt bibliotek av klasser
- En programmeringsmodell
- En gemensam motor för att köra kod i många olika språk
 - C#
 - Visual Basic .NET
 - C++
 - F# •
 - ... och många andra

.NET - Historik

- Microsoft startade utvecklingen av .NET Framework i slutet av 1990-talet under namnet Next Generation Windows Services (NGWS).
- I slutet av 2001 släpptes de första betaversionerna av .NET 1.0.
- Den första versionen av .NET Framework släpptes den 13 februari 2002 .
- Sedan den första versionen har Microsoft släppt nio fler uppgraderingar för .NET Framework, varav sju har släppts tillsammans med en ny version av Visual Studio.
- .NET Framework-familjen innehåller också två versioner för mobil eller inbäddade enheter. En reducerad version av ramverket, .NET Compact Framework, är tillgänglig på Windows CE-plattformar, inklusive Windows Mobile-enheter som smartphones. Dessutom är .NET Micro Framework inriktat på hård resurskontrollerade enheter.
- .NET Framework 4.8 är den sista versionen av .NET Framework.
- I framtiden hamnar all ny kod i .NET Core-plattformen som .NET 5, i slutet av 2020.

.NET – Vad är det?

- .NET (uttalas dot net) är ett ramverk som ger riktlinjer för hur man programmerar, som
- Kan användas för att utveckla ett brett spektrum av applikationer.
 - Det kan vara Webb till mobil till Windows-baserade applikationer.
 - .NET framework kan fungera med flera programmeringsspråk som C#, VB.NET, C ++ och F#.
 - .NET fungerar numera på en stor rad andra plattformar, såsom Mac, Linx, Android m fl
- .NET innehåller ett stort antal funktioner, eller klasser som löser en rad olika problem.

.NET, .NET Framework och .NET Core

- I början fanns .NET Framework •
 - Webbplatser, tjänster, skrivbordsapplikationer mm
 - Endast på Windows
- Sedan kom .NET Core
 - Webbplatser, tjänster, konsollapplikationer
 - På flera plattformar: Windows, Linux, MacOS •
 - Open Source, fritt tillgängligt
- .NET från november 2020
- .NET Core blev .NET version 5
- Nu kör vi .NET 6
 - Läs mer på <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>

.NET Standard

- .NET körs på flera plattformar
- .NET Standard är en specifikation för hur .NET ska fungera •
- .NET Standard gör det möjligt att skriva kod en gång •
 - Och köra på olika maskiner (Windows, Linux, Mac)
- [.NET Standard | Common APIs across all .NET implementations \(microsoft.com\)](#)

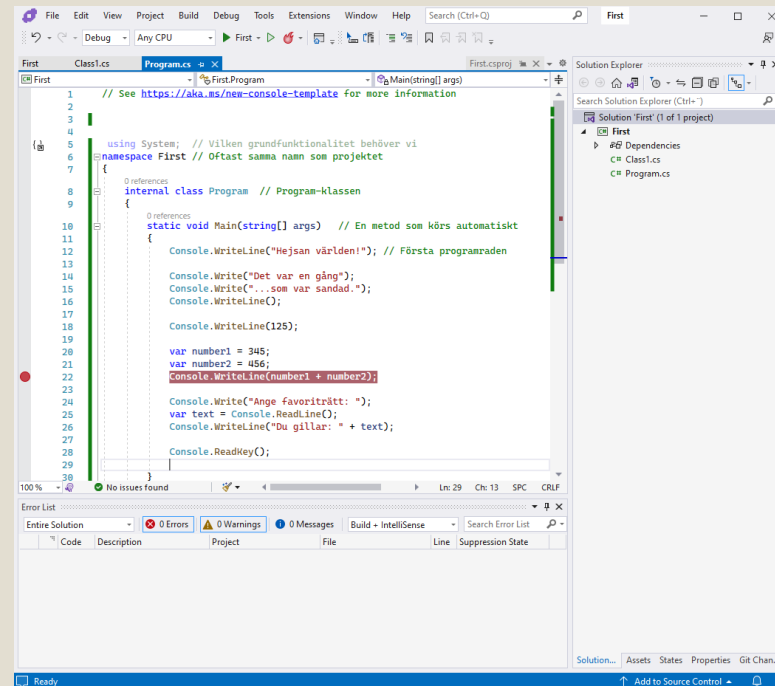
Visual Studio

- En komplett utvecklingsmiljö
- En modern IDE – Integrated Development Environment
- Verktyg för att
 - Skriva kod
 - Bygga användargränssnitt
 - Kompilera kod
 - Köra/testa/avlusa applikationer
 - Hantera filer i ett projekt

Visual Studio fördelar

- Ett enda verktyg för
 - Skriva kod i flera språk (C#, F#, VB,...)
 - Flera tekniker (Web, skrivbord, konsoll, mobil...)
 - För olika plattformar (Web, Windows, Linux, Mac,...)
 - Full integration av de flesta utvecklingsaktiviteter
 - Kodning, kompilering, test, debug, driftsättning, versionshantering, ...
 - Väldigt lätt att använda

Visual Studio Exempel

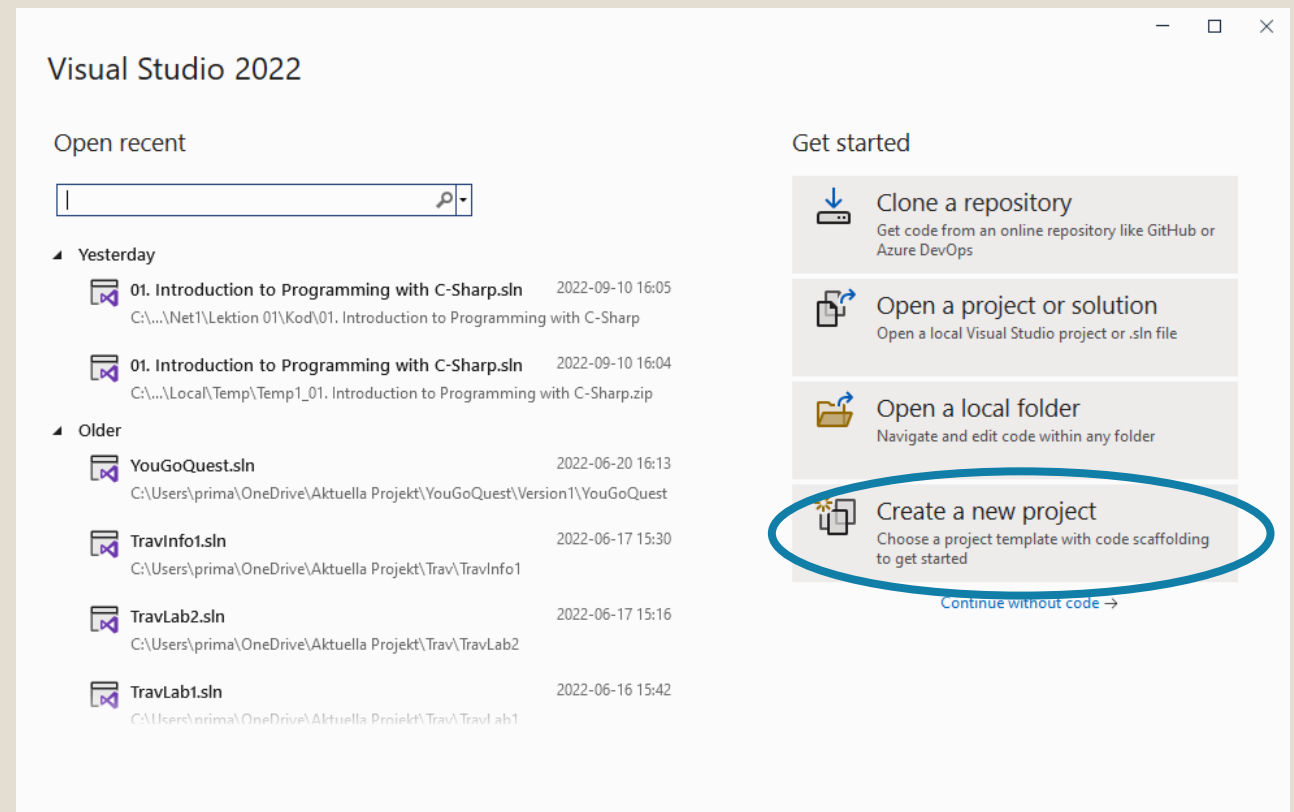


Visual Studio - Installera

- Den version vi ska använda är Visual Studio Community 2022, som är gratis.
- [Visual Studio 2022 IDE - Programming Tool for Software Developers \(microsoft.com\)](#)
 - Detta laddar ner Visual Studio Installer, och därifrån installeras de delar du vill ha.

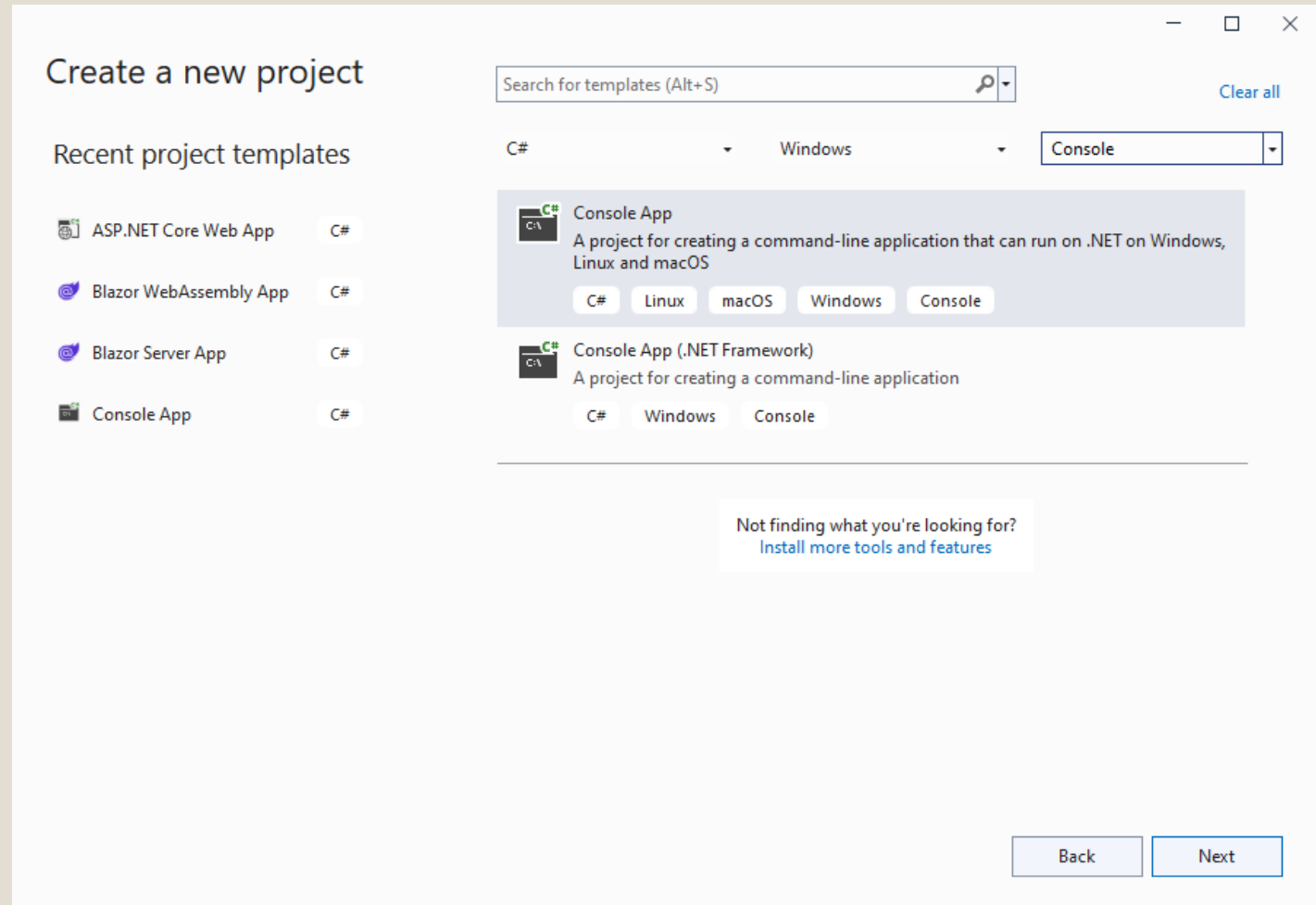
Skapa nytt projekt

- Vi kommer jobba med s k console-project, ett tag.
- Starta Visual Studio 2022
- Du möts av den här skärmen:
- Klicka på Create a new project



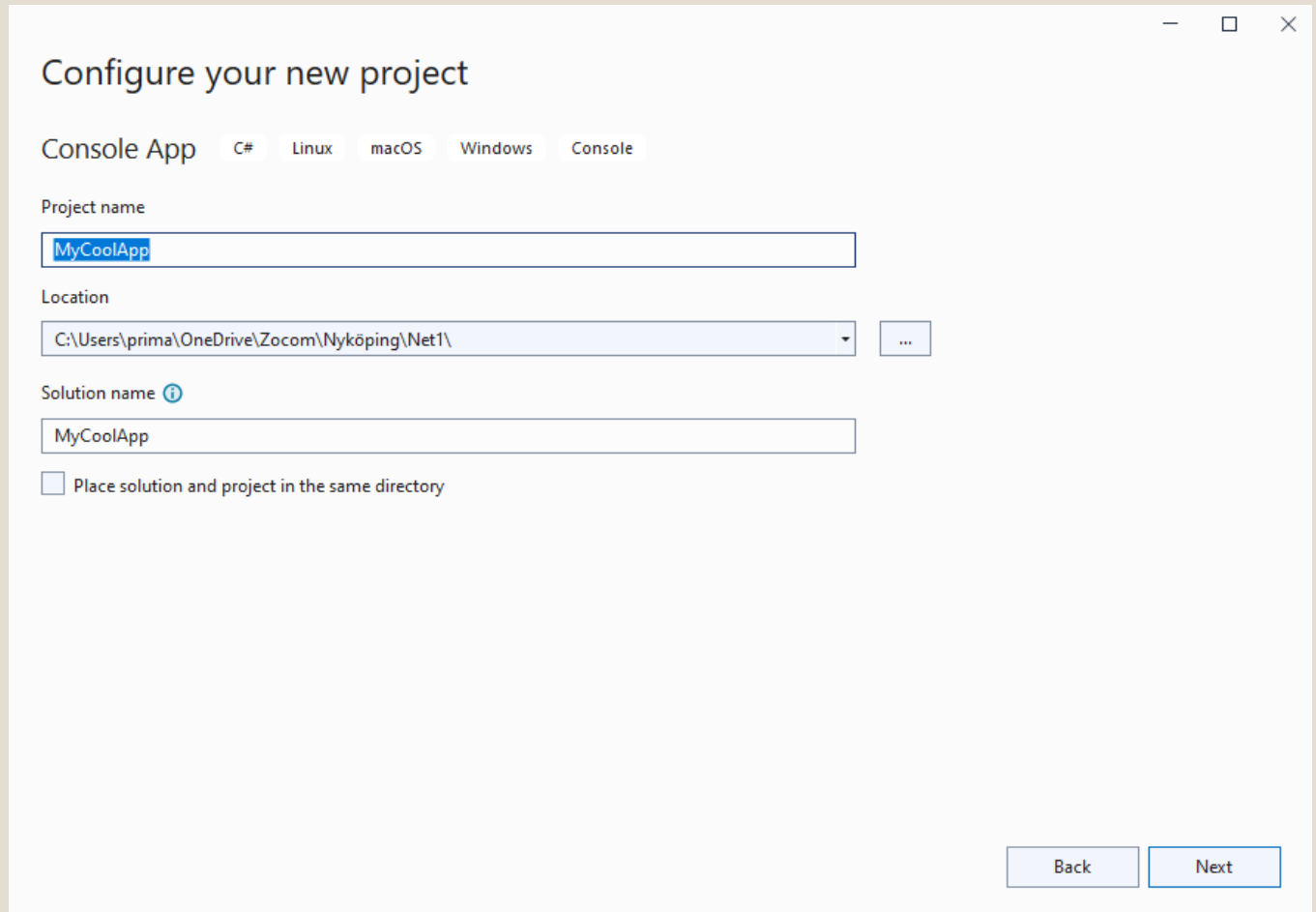
Skapa nytt project forts.

- Leta upp “Console App”
 - Välj i dropdowns:
 - C#
 - Windows
 - Console



Skapa nytt project forts.

- Namnge projektet.
- Välj var det ska sparas:



Configure your new project

Console App C# Linux macOS Windows Console

Project name

MyCoolApp

Location

C:\Users\prima\OneDrive\Zocom\Nyköping\Net1\

Solution name ⓘ

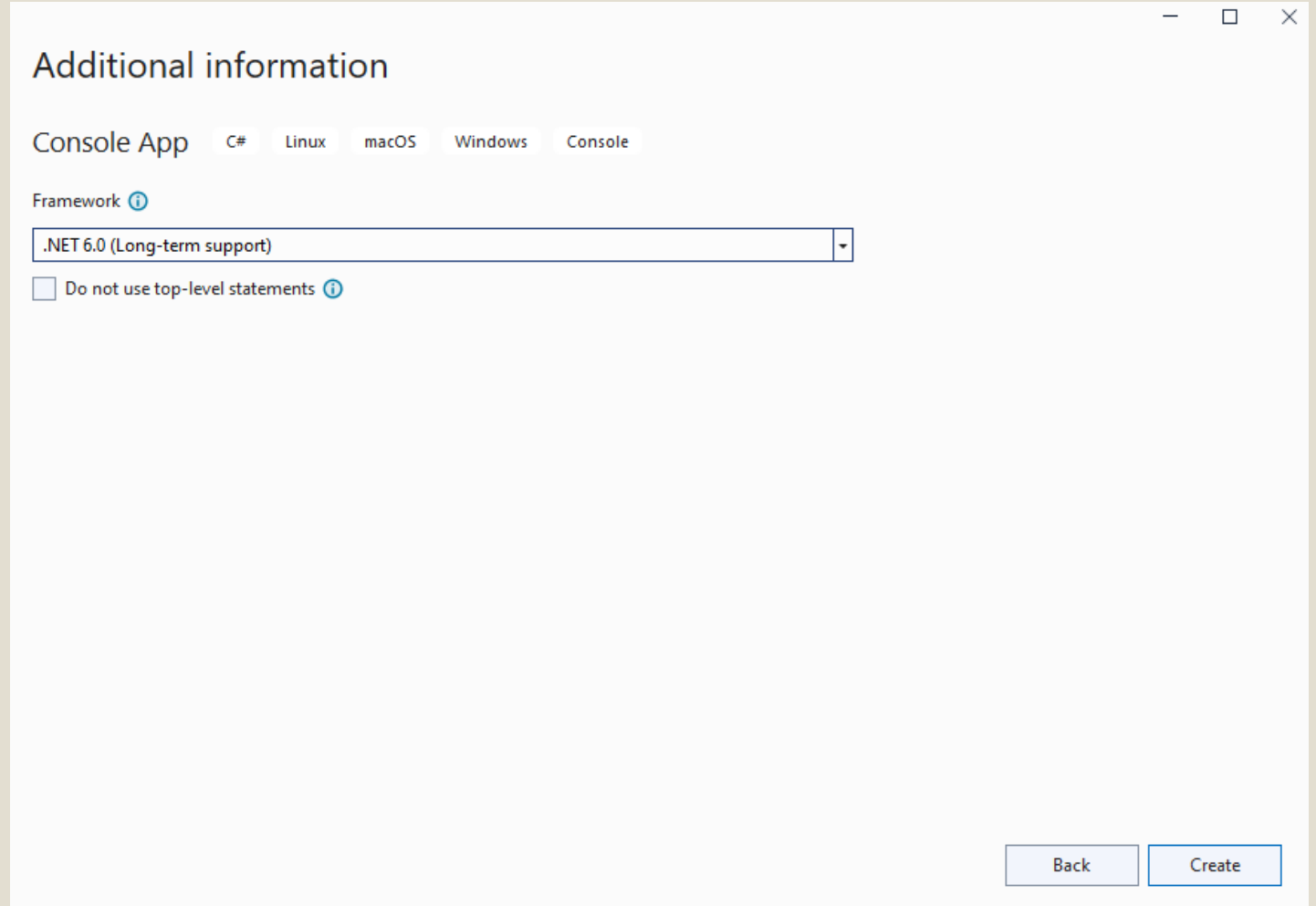
MyCoolApp

☐ Place solution and project in the same directory

Back Next

Skapa nytt project forts.

- Välj .NET 6.0



The screenshot shows the 'Additional information' window in Visual Studio. At the top, there are tabs for 'Console App', 'C#', 'Linux', 'macOS', 'Windows', and 'Console'. The 'Console App' tab is selected. Below the tabs, the 'Framework' is set to '.NET 6.0 (Long-term support)'. There is a checkbox labeled 'Do not use top-level statements' which is currently unchecked. At the bottom right, there are 'Back' and 'Create' buttons.

Additional information

Console App C# Linux macOS Windows Console

Framework ⓘ

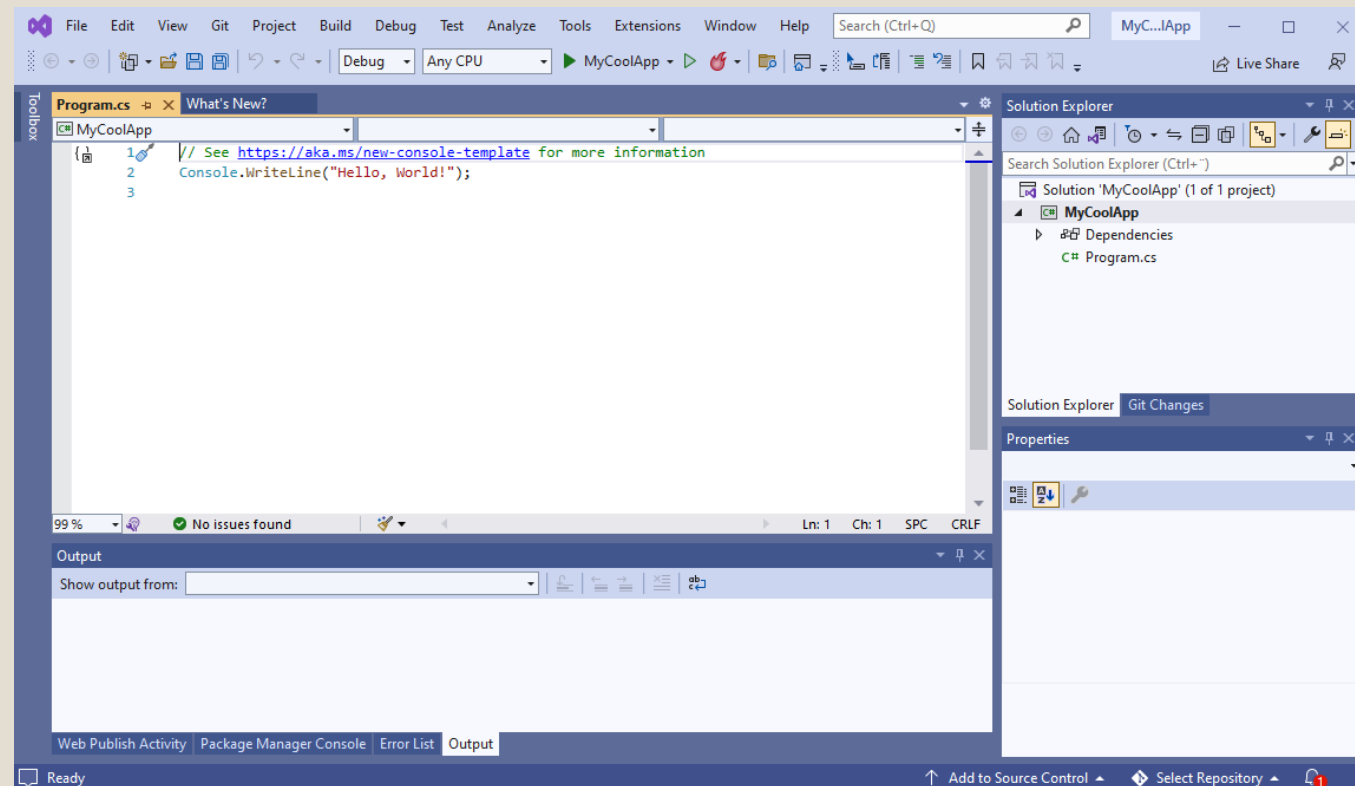
.NET 6.0 (Long-term support) ▼

☐ Do not use top-level statements ⓘ

Back Create

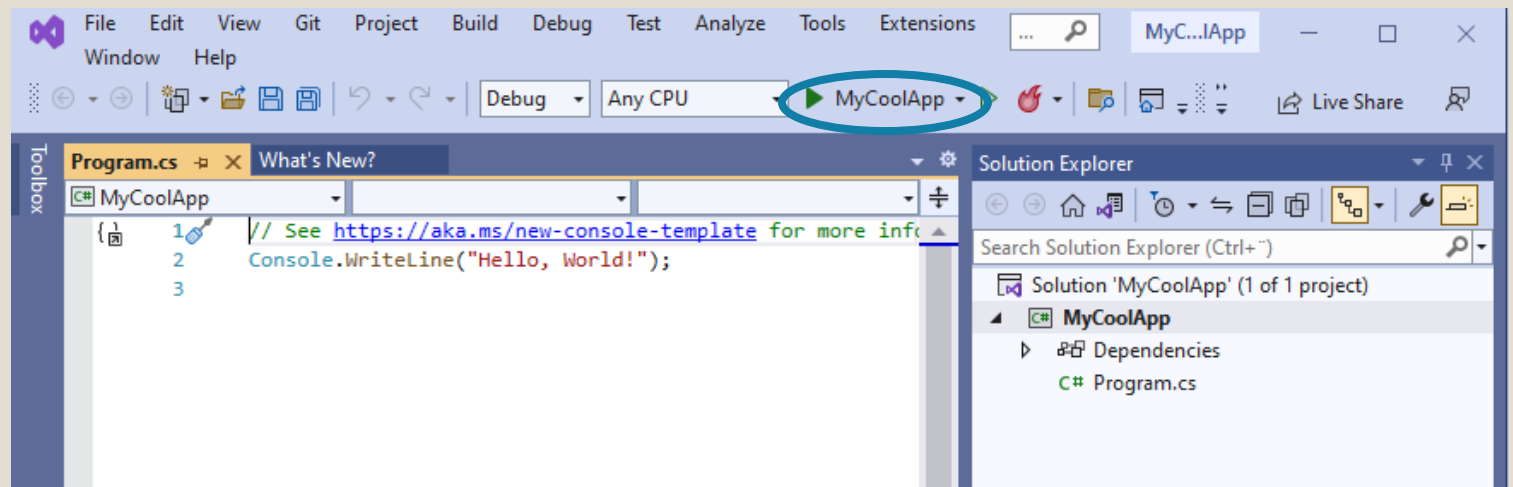
Klart!

- Utseendet ser olika ut beroende på vilket Tema man valt och hur fönstren är placerade, men ungefär såhär kan det se ut.



Viktigaste funktionerna

- Kompilera källkod
 - Att kompilera kod innefattar:
 - Syntaxkontroll
 - Kontroll av typsäkerhet
 - Översättning av källkoden till lågnivåspråk
 - Skapa körbara filer (assemblies)
 - Du kan starta kompileringen, och köra igång appen genom att
 - Välja Debug -> Start
 - Trycka på [F5] eller [Ctrl+F5]
 - Eller klicka här:

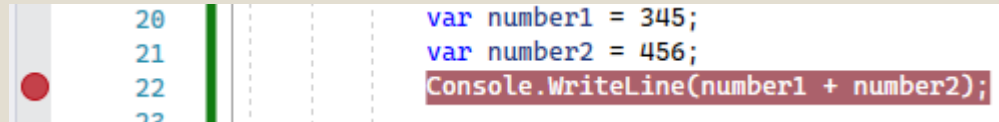


Debugga (avlusa) koden

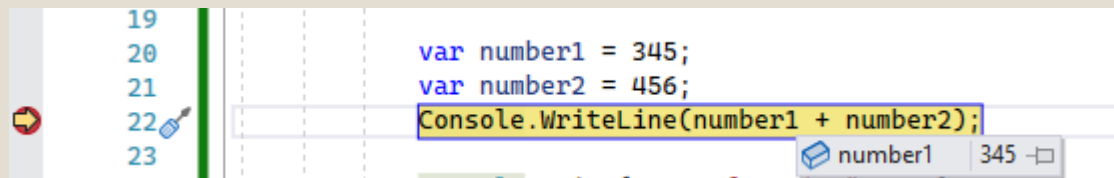
- Att avlusa (debug) koden innefattar:
 - Upptäcka ett fel
 - Hitta var i koden felet uppstår
 - Fixa koden
 - Testa för att se att felet är borta
 - Och för att se till att inga nya fel uppstod
 - Detta är en iterativ och kontinuerlig process

Debugga I Visual Studio

- Visual Studio har en inbyggd debugger
- Den ger:
 - Brytpunkter (breakpoints)
 - Klicka till vänster om kodraden, där du vill att kodkörningen ska stanna (stannar precis före den kör just den raden)



- Möjlighet att se vilken väg man har gått igenom koden
 - Klicka på Continue, för att köra vidare i koden
 - Tryck F11 för att stega dig igenom koden
- Möjlighet att inspektera variabler medan programmet körs



å

Konkatenering

- En vanlig sak man gör är att sätta ihop texter/strängar.
- Exempel
 - `Var text1 = "Hej"`
 - `Var text2 = "Hopp"`
 - `Console.WriteLine(text1 + " " + text2)`
 - `Console.WriteLine($"{text1} {text2}")`
 - `Console.WriteLine(String.Format("{0} {1}", text1, text2));`

Datetime

- `Console.WriteLine("Idag är det " + DateTime.Now); // Dagens datum och tid`
- `Console.WriteLine("Året är " + DateTime.Now.Year); // Nuvarande år`

Översätta text till siffror

- `int number = int.Parse(text);`

Hello world I .NET 6 C#

```
Console.WriteLine("Hejsan världen");
```

Code Along

- Äntligen blir det lite kodning!
- Skapa Konsolapplikation
- Titta runt i Visual Studio
 - Anpassa gränssnittet
 - Filstruktur
 - Dependencies
- Några enklare första kodrader

Övningar

- Bekanta dig med Visual Studio, Console-appar och C#
 - Skapa, kompilera och kör en "Hello C#" konsolapplikation
 - Ändra applikationen så att den skriver ut en annan text, som du själv matar in.
 - Skapa ett program som skriver ut talen 9, 155 och 312, samt summan av talen.
 - Skapa en konsolapplikation som skriver ut aktuellt datum och klockslag i formatet "Idag är det 2022-09-12 13:51:40"
 - Skapa en konsolapplikation som beräknar och skriver ut summan av två valfria tal, som matas in av användaren.
 - Skapa ett program som läser ditt födelseår från konsolen och skriver ut hur gammal du är nu.
 - Sagan: Skapa en applikation som bygger en saga enligt följande upplägg:
 - Det var en gång en <ålder> år gammal <djursort> som hette <namn>. En dag var <namn> ute på en promenad i skogen, och mötte en stor varg. Vargen bet av ett ben. <namn> sprang snabbt hem på sina <antalben - 1>. Så var sagan slut.

Länkar

- [C# docs - get started, tutorials, reference. | Microsoft Docs](#)
- [C# Tutorial \(C Sharp\) \(w3schools.com\)](#)
- [What is .NET Framework? A software development framework. \(microsoft.com\)](#)
- [.NET Standard | Common APIs across all .NET implementations \(microsoft.com\)](#)