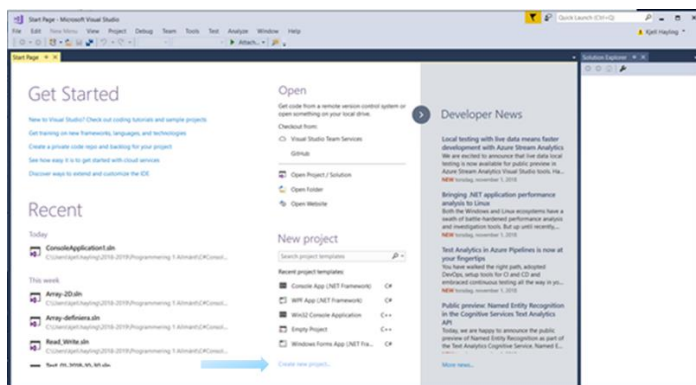


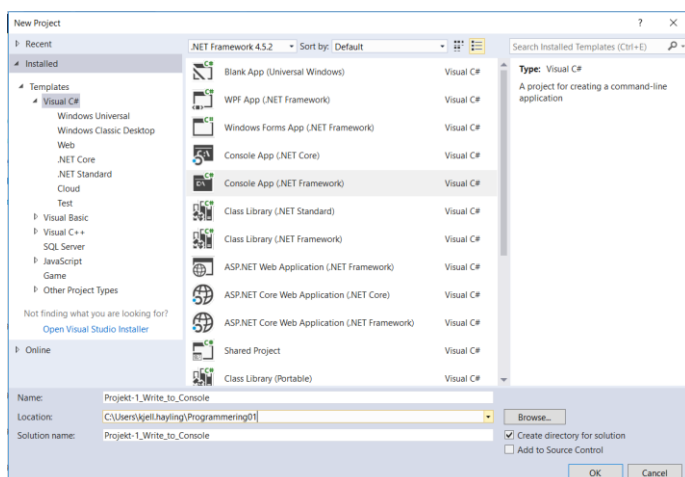
Programmering C#.

Starta ett projekt i Visual Studio.

1. Starta Visual Studio. Start sidan kommer att se ut som bild 1, men med andra program under "Recent" beroende på vilka program som använts tidigare.
2. Tryck på **Create new project** under **New project**



3. I den nya menyn markera Console App (.NET Framework) om den inte redan är markerad.



4. I de tre nedersta rutorna anges vad projektet skall heta (Name:) och i vilket bibliotek projektet skall läggas (Location:). I detta exempel heter projektet Projekt-1_Write_to_Console och har lagts i biblioteket C:\Users\kjell.hayling\Programmering01. Per automatik kommer "Solution name:" att bli samma som "Name:"
5. Tryck på OK och ett projekt med namnet "Projekt_1_Write_to_Console" har skapats.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Projekt_1_Write_to_Console
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13        }
14    }
15 }
16

```

Rad 1 – 5 skapar (potentiella) länkar till olika bibliotek som innehåller funktioner och metoder som kan användas när program skapas. De bibliotek som kommer att behövas i programmet kommer att markeras med mörkare text.

Rad 6 är en blank rad och har ingen annan funktion än att göra programkoden tydligare och enklare att förstå.

Rad 7 `namespace`, namnrymd på svenska, gör att namnen på variabler, metoder (funktioner) och klasser är unika inom projektet. (Ett bra, ofta använt, exempel är e-postadresser, t.ex. e-postadressen kalle.kula@lbs.se är unik då det endast finns en Kalle Kula på lbs.se, men det kan finnas andra Kalle Kula på andra e-postadresser, t.ex. kalle.kula@elevarbetensys.se.)

Rad 8 `{` indikerar att här börjar programmet. Denna `{` hänger samman med `}` på rad 15 och indikerar vad som omfattas av programmet "Projekt_1_Write_to_Console".

Rad 9 deklarerar klassen att programmet

Rad 10 `{` indikerar att . denna hänger samman med `}` på rad 14.

Rad 11 deklarerar metoden att programmet

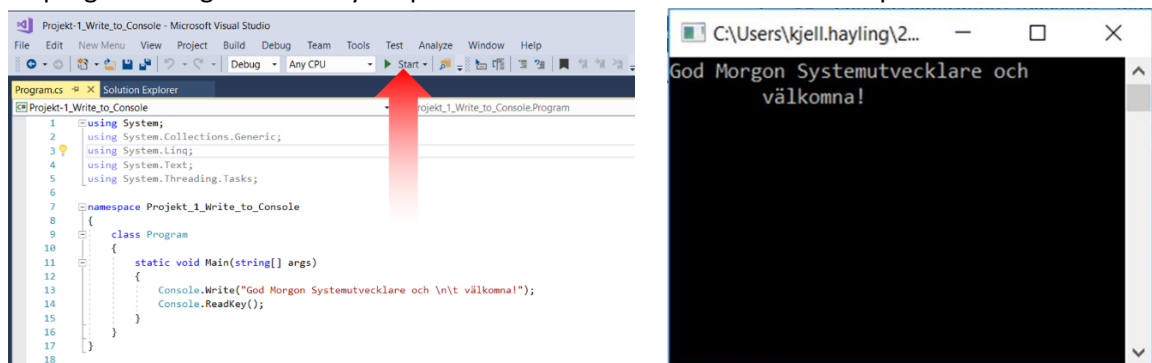
Rad 12 `{` indikerar att här börjar programkoden. Denna hänger samman med `}` på rad 13.

Rad 13 `}` avslutar metoden

Rad 14 `}` avslutar klassen

Rad 15 `}` avslutar programmet??

6. Sätt pekaren bakom `{` på rad 12 och tryck ENTER för att skapa en ny rad
7. Skriv in följande två rader,
 - `Console.WriteLine("God Morgon Systemutvecklare och \n\t välkomna!");`
 - `Console.ReadKey();`
8. Kör programmet genom att trycka på Start. Resultatet skrivs ut i konsolen på skärmen.



9. Avsluta programmet genom att trycka någon tangent.

Den nya rad 13 - `Console.WriteLine("God Morgon Systemutvecklare och \n\t välkomna!");` skriver till konsolen

```
God Morgon Systemutvecklare och
    Välkomna!
```

`\n` skapar en ny rad vid utskriften

`\t` skapar en tabulering (tabbar).

Den nya rad 14 - `Console.ReadKey();` gör att datorn väntar på en tangenttryckning för att fortsätta.

Nedan ges källkoden, som helt eller delvis kan kopieras och "klistras in" i Visual Studio.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("God Morgon Systemutvecklare och \n\t välkomna!");
            Console.ReadKey();
        }
    }
}
```

När vi tittat lite närmare på programkoden ser vi att raderna 2 – 5 har skuggad text. Detta beror på att dessa bibliotek (klasser) som anropas automatiskt inte behövs av det program som vi skapat. Dessa kan i detta fall tas bort.

```
1 using System;
2
3 namespace Projekt_1_Write_to_Console
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("God Morgon Systemutvecklare och \n\t välkomna!");
10            Console.ReadKey();
11        }
12    }
13 }
```

Uppgifter. Skapa projekt i Visual Studio med de programkoder som laddats upp på Google Classroom.

- Array-1.txt
- Array-2D.txt
- Lista-1.txt
- Lista-2.txt
- Slumptal.txt
- Metod-1.txt

- Metod-2.txt
- Metod-3.txt
- RandomPassWord-1.txt
- RandomPassWord-Class.txt

Övningsuppgifter.

1. Skapa ett program som beräknar hur många dagar det tar att bli miljonär om ett kapital startar med 1 öre och växer genom att dubblas för varje dag
 - Dag 1 - 1 öre
 - Dag 2 - 2 öre
 - Dag 3 - 4 öre
 - Dag 4 - 8 öre
 - osv.
2. Skapa ett program som beräknar kostnaden per månad för ett lån med amortering (avbetalning) under en viss period, t.ex. 7 år, samt med en räntesats om t.ex. 5%.

Avancerad uppgift

3. Skapa ett program som beräknar kostnaden per månad för ett annuitetslån (samma kostnad varje månad (räntekostnaden är högre i början av avbetalningarna) under samma period som i uppgift 2, t.ex. 7 år, samt med en räntesats om t.ex. 5%. (Testa din beräkning med någon beräkningsfunktion , t.ex. <http://lan-info.com/kalkylator-annuitetslan.php>)

Sammanfattning av variabler och operatorer

Operatorer

<u>Operator</u>	<u>Betyder</u>	<u>Exempel</u>
*	gånger (multiplikation)	$7 * 5 = 35$
/	delat med (division)	$7 / 5 = 1$
% (modulo)	heltals rest	$7 \% 5 = 2$
+	plus (addition)	$7 + 5 = 12$
-	minus (subtraktion)	$7 - 5 = 2$

Jämförelseoperatorer

<u>Operator</u>	<u>Betyder</u>	<u>Exempel</u>
<	Mindre än	if (a<20)
>	Större än	if (b>100)
==	Lika med	if (produkt==500)
!=	Skiljt från (inte lika med)	if (x!=28)
<=	Mindre än eller lika med	if (a<=2000)
>=	Större än eller lika med	if (b>=12)

Logiska operatorer

<u>Operator</u>	<u>Betyder</u>	<u>Exempel</u>
&&	och	if ((a>20) && (a<30))
	eller	if ((a==0) (b<0))
!	inte	if (!(a>0))

Variabeltyper

short: heltal (positiva och negativa)	2 byte
int: heltal (positiva och negativa)	4 byte
long: heltal (positiva och negativa)	8 byte
float: flyttal, reellt tal (decimaltal)	4 byte
double: flyttal, reellt tal(decimaltal)	8 byte
char: ett tecken	2 byte
string: textsträngar (ett antal tecken)	2 byte/tecken
bool: sant eller falskt	1 byte

Det går att använda alla svenska bokstäver, siffror, dollartecken och understreck. Observera att det är skillnad mellan stora och små bokstäver. **MEN DET GÅR INTE att börja variabelnamnet med en siffra.**