

TDDE71

Kursupplägg och Intro till C++

Eric Ekström & Klas Arvidsson

Institutionen för datavetenskap

Handout version

1 Kursinformation

2 Grunder i C++

Hello World

Kompilering

Utströmmar

Variabler

Inströmmar

Villkor

TDDE71 - Syfte

Lära sig

- grunderna i C++
- objektorientering i C++
- testdriven utveckling
- datastrukturer och algoritmer
- att jobba i större programmeringsprojekt

TDDE71 - Kursinnehåll

- 8 Föreläsningar
- 4 Laborationer (egen tid + 11 schemalagda pass)
- 4 Seminarier
- Dugga
- Datastrukturer och algoritmer genom OpenDSA (egen tid)
- Projekt (HT2)

All information finns på kurshemsidan:

<https://www.ida.liu.se/~TDDE71/>

TDDE71 - Examination

- **LAB1** - 3hp Laborationer
- **DAK2** - 0.5hp Datordugga
- **PRA1** - 3hp Projekt
- **UPG1** - 1.5hp Datorbaserade inlämningsuppgifter (OpenDSA)

För betyg 3 i kursen ska alla fyra moment vara godkända.

Högre betyg fås genom att samla bonuspoäng under kursens gång.

TDDE71 - Högre betyg

- Betyg 3: alla moment godkända
- Betyg 4: alla moment godkända + 20 poäng
- Betyg 5: alla moment godkända + 40 poäng

Poäng samlas genom att ligga i fas i kursen och göra extrauppgifter.

Fullständig lista med möjliga poäng finns på kurshemsidan.

TDDE73 - Kursinnehåll

HT1:

- 6 Föreläsningar
- 4 Laborationer (egen tid + 11 schemalagda pass)
- 4 Seminarier

HT2:

- 4 Föreläsningar
- Inlämningsuppgifter

TDDE73 - Examination

- **LAB1** - 3hp Laborationer
- **UPG1** - 3hp Inlämningsuppgifter

För betyg 3 i kursen ska båda moment vara godkända.

TDDE73 - Högre betyg

- Betyg 3: alla moment godkända
- Betyg 4: 20 poäng (HT1) **eller** 5 stjärnmärkta uppgifter (HT2).
- Betyg 5: 20 poäng (HT1) **och** 5 stjärnmärkta uppgifter (HT2).

TDDE71: Laborationer

- Arbeta i par
- Anmälan i Webreg innan första labbpasset
- Sendlab för kodinlämning
- Deadlines i schemat
- Samla bonus genom att ligga i fas

TDDE71: Personal

- Examinator - Klas Arvidsson (klas.arvidsson@liu.se)
- Kursledare - Eric Ekström (eric.ekstrom@liu.se)
- Assistenten - Jesper, Louise, Niels, Nils (Nisse), Klas, Eric
- Kontaktuppgifter på hemsidan

TDDE73 (DALG):

- Kursledare - Christoffer Holm (christoffer.holm@liu.se)

TDDE71: Utvärdering och Återkoppling

Evaluat 2024 (svarsfrekvens 34%)

Helkursbetyg: 3.14 (från 4.64 år 2023)

Studentkommentarer:

- OpenDSA känns bortkopplat från resten av kursen
- Seminarierna var bra
- Önskar fler schemalagda pass under hela kursen
- Duggan var katastrof

TDDE71: Utvärdering och Återkoppling

Ändringar tills i år:

- Skrivtiden förlängs (2→3 timmar)
- Duggauppgiften byts ut
En liten uppgift från varje labb
Exempel på kurshemsidan
- Projektspår tas bort
- Duggadatum ändras

1 Kursinformation

2 Grunder i C++

Hello World

Kompilering

Utströmmar

Variabler

Inströmmar

Villkor

Ett enkelt program i C++

Python

```
print("Hello World")
```

C++

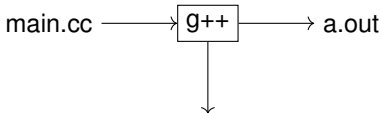
```
#include <iostream>

int main()
{
    std::cout << "Hello World" << std::endl;
    return 0;
}
```

- Instruktioner avslutas med ;
- Alla instruktioner skrivs inom main()
- Indentering i Python == Måsvingar i C++
- #include <iostream> låter oss använda std::cout

Kompilering

C++ är ett kompilerat språk. Vi kommer använda kompilatorn g++.



```
main.cc: In function 'int main()':  
main.cc:8:9: error: cannot convert 'string' to 'int' in assignment  
      8 |         x = num;  
        |         ^  
        |         string
```

- En kompilator är ett speciellt program
- Den översätter källkodsfiler till exikverbara filer
- En exikverbar fil innehåller maskinkod som datorn kan köra

Kompilering

Vi kompilerar koden från terminalen

```
$ g++ main.cc  
$ ls  
a.out main.cc
```

Om inget skrivs ut har kompileringen lyckats

Programmet kan då köras med

```
$ ./a.out
```

Kompilering

Vi vill kompilera med extra kontroller

```
$ g++ -std=c++17 -Wall -Wextra -pedantic -Werror main.cc
```

Långt att skriva? Skapa ett alias!

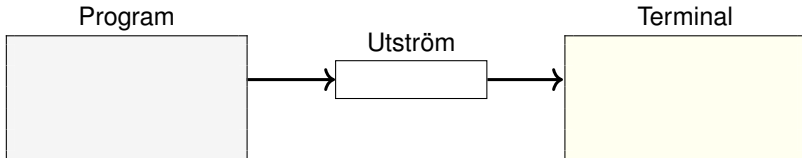
```
alias w++17='g++ -std=c++17 -Wall -Wextra -pedantic'  
alias e++17='g++ -std=c++17 -Wall -Wextra -pedantic -Werror'
```

Dessa två kommer finnas i tentamiljö.

- `-Wall -Wextra` lägger till fler varningar från kompilatorn, vilket hjälper oss att programmera bättre.
- `-Werror` ändrar alla varningar till fel.
- `-std=c++17` låter oss välja versionen av C++. C++17 är den vi använder.

Utströmmar

Vi kan kommunicera med terminalen genom en utmatningsbuffer.



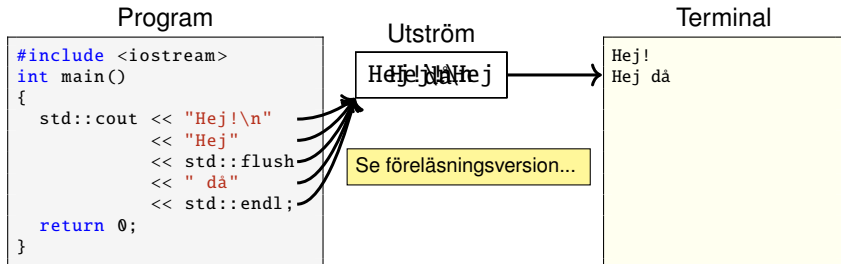
I C++ implementeras detta som “strömmar”.

```
std::cout << "Hej!" << std::endl;
```

Utströmmar

- `std::cout` används för att kommunicera med utströmmen.
- `operator<<` skickar data till strömmen.
- `std::flush` tömmer strömmen till terminalen.
- `std::endl` lägger en nyrad i strömmen och gör en `std::flush`.

Utströmmar



Handout-slides komprimerar animationer till 1 slide.

Utströmmar - Formaterad utskrift

```
#include <iostream>
#include <iomanip>

int main()
{
    double const pi { 3.141592 };
    std::cout << std::setfill('.') << std::left
               << std::setprecision(3)
               << std::setw(6)
               << pi << std::endl;

    std::cout << std::fixed << pi << std::endl;
    return 0;
}
```

```
3.14..
3.142
```

Utströmmar

Vi kan formatera utskriften m.h.a funktioner från `<iostream>` och `<iomanip>`.

- `setprecision(N)` - Använd N värdesiffror
- `fixed` - Ändrar `setprecision` till att ange antal decimaler istället.
- `setw(N)` - Använd minst N tecken för nästa utskrift
- `setfill(C)` - Använd tecknet 'C' som utfyllnadstecken
- `left` - Vänsterjustera utskriften (inom `setw`)
- `right` - Högerjustera utskriften (inom `setw`)

Ett *lite* mer komplicerat program i C++

Python

```
number = input("Vilket heltal är bäst? ")

if number == 76:
    print("Korrekt")
else:
    print("Fel")
```

C++

```
#include <iostream>
int main()
{
    std::cout << "Vilket heltal är bäst? "
                << std::flush;
    int number {};
    std::cin >> number;

    if (number == 76)
    {
        std::cout << "Korrekt" << std::endl;
    }
    else
    {
        std::cout << "Fel" << std::endl;
    }
    return 0;
}
```


Datatyper och Variabler

För att spara data under programkörning behöver vi lista ut ett par saker:

- Var i minnet placerar vi data?
- Hur hittar vi den datan senare?
- Hur mycket plats behövs i minnet?
- Hur tolkas datan?

Datatyper och Variabler

Variabler i C++ har alltid

- ett unikt namn (a-Z, 0-9, _)
- ett värde (även om vi inte har angett något!)
- en specifik datatyp. De vanligaste är

int	Heltal	-7	2147483647
float, double	Flyttal	14.7	0.99f
char	Tecken	'j'	'\n'
std::string	Text	"Hej jag heter"	
bool	Sanningsvärde	true	false

En variabel i C++ kan aldrig byta typ

Datatyper och Variabler

```
#include <string>

int main()
{
    int x { 4 };
    char c { 'e' };
    float const pi { 3.14 };
    std::string erics_passwd { "c++ärbäst" };

    x = 10;
    pi = 9;

    return 0;
}
```

KOMPILERAR EJ!

- Datatypen måste alltid anges när vi skapar variabler.
- Vi initialiserar variabler med {}

int
x: 10

char
c: 'e'

float
pi: 3.14

Datayper och Variabler - std::string

Strängar i C++ är inte en inbyggd datatyp.

- `std::string` är definierad i `<string>` som måste inkluderas.
- Strängar har trevlig inbyggd funktionalitet vi kan utnyttja.

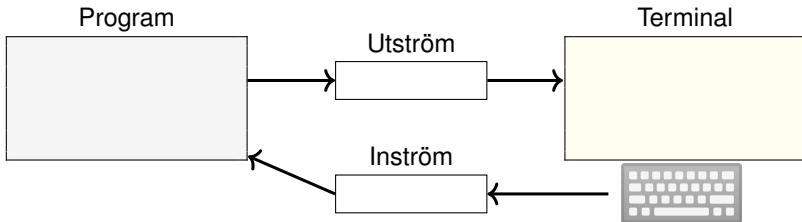
```
#include <iostream>
#include <string>

int main()
{
    std::string str {"hello"};
    std::cout << str      << '\n'
              << str.size() << '\n'
              << str.front() << std::endl;

    return 0;
}
```

Inströmmar

Det finns även en inström från terminalen till programmet som vi kan kommunicera genom.



Exempelvis

```
std::cin >> x;
```

Inströmmar

- `std::cin` kan vi använda för att kommunicera med inströmmen.
- med operator `>>` läser vi data från strömmen.
- Vad kan stå på höger sida om `>>`?
Är detta giltigt?

```
std::cin >> 5;
```

Inströmmar

temp: double
24.5 stl: int
20

Program

```
#include <iostream>
int main()
{
    std::cout << "Mata in
    rumstemperatur
    och rumstl:"
    << std::flush;

    double temp {};
    int stl {};
    std::cin >> temp;
    std::cin >> stl;
    return 0;
}
```

Utström

Mata in rumstemperatur
och rumstl:

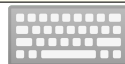
Oj! Se föreläsningsvarianten...

Inström

24.5 20\n\n

Terminal

Mata in rumstemperatur
och rumstl: 24.5 20



Villkor

```
int main()
{
    std::cout << "Vad är temperaturen? "; // Ingen flush??
    int temp {};
    std::cin >> temp;
    if (temp >= 40)
    {
        std::cout << "Är du i en bastu?" << std::endl;
    }
    return 0;
}
```


Villkor

```
int main()
{
    std::cout << "Vad är temperaturen? "; // Ingen flush??
    int temp {};
    std::cin >> temp;
    if (temp >= 40)
    {
        std::cout << "Är du i en bastu?" << std::endl;
    }
    else
    {
        std::cout << "Låter alldeles lagom" << std::endl;
    }
    return 0;
}
```

Villkor

```
int main()
{
    std::cout << "Vad är temperaturen? "; // Ingen flush??
    int temp {};
    std::cin >> temp;
    if (temp >= 40)
    {
        std::cout << "Är du i en bastu?" << std::endl;
    }
    else if (temp < 15)
    {
        std::cout << "Maila studentbostäder" << std::endl;
    }
    else
    {
        std::cout << "Låter alldeles lagom" << std::endl;
    }
    return 0;
}
```

Misslyckade inläsningar

Kan vi kontrollera om en inläsning gått bra?

```
int temp {};  
if ( std::cin >> temp )  
{  
    std::cout << "Du matade in temperaturen " << temp << std::endl;  
}  
else  
{  
    std::cout << "Tyvärr, du matade inte in ett giltigt heltal"  
              << std::endl;  
}
```

Vad gör vi nu...

Mer om inläsning

- `std::cin.clear();`
Nollställer felflaggan så vi kan fortsätta läsa.
- `std::cin.ignore(5, '\n');`
Rensar strömmen på max 5 tecken, eller stannar vid första nyradstecknet.
- `std::getline(std::cin, str, '\n');`
Läser en hel rad tecken (fram till nyradstecknet) och sparar dessa i variabeln `str`.

Vad har vi gått igenom?

- Kompilering
- Utmatning
- Variabler
- Inmatning
- Villkor

Resurser och Informationssökning

Hur hittar man information om C++?

- Givet material i kursen (se kurshemsidan)
- Rekommenderad bok om C++
- Fråga kurspersonal (i sal eller på mail)
- Sök på `cppreference.com`

Vanskliga alternativ (ditt mål är att *lära* dig C++, *inte* att producera en labblösning)

- Fråga kompisar
- Google? chatGPT?

Om du ställer frågan “varför...?” inte “hur...?”

Finn fem fel!

```
#include <string>
int main() {
    std::cout << "Skriv in två heltal: " << std::endl;
    int x; int y {};
    std::cin << x << y;
    if ( std::cin )
    {
        if x < y
            std::cout << x << " är mindre än" << y;
    }
    else
    {
        std::cout << "inläsningen misslyckades" << std::endl;
    }
}
```

Finn *minst nio* fel!

- string inkluderas istället för iostream
- x deklarerar utan måsvingar
- pilarna i strömoperatörn för std::cin går åt fel håll
- if-satsen saknar parenteser och måsvingar
- det saknas en std::flush på andra utskriften
- två satser ligger på samma rad
- det saknas en return 0 i slutet av main
- koden är för kompakt skriven
- med mera...

www.ida.liu.se/~TDDE71/