



Norges teknisk–naturvitenskapelige  
universitet  
Institutt for datateknologi og  
informatikk

TDT4102 Prosedyre-  
og objektorientert  
programmering  
Vår 2019

Øving 0 for Windows

## Frist: Ingen (frivillig øving)

### Mål for denne øvingen:

- Bli kjent med programmeringsverktøy
- Lage et første program med Visual Studio
- Kunne laste ned og kjøre eksempelprogram fra forelesningene med Visual Studio
- Lage et første program kun med teksteditor og kompilator

Denne øvingen er en veiledning i å installere en programmeringsomgivelse slik at du kan skrive, redigere, kompilere, debugge og kjøre et C++ program. **Det er nødvendig å gjennomføre og mestre det meste av det som gjennomgås i denne øvingen for å kunne utføre de obligatoriske øvingene.**

Vi vil våren 2019 benytte verktøyet Visual Studio 2017 Community Edition som er gratis. Dersom du bruker en tidligere versjon av programmet er det sannsynlig at deler av den følgende beskrivelsen ikke stemmer. Foreleserne vil bruke dette verktøyet, som vi ofte forkorter til bare MS-VS. Vi kan kun hjelpe med installering av bibliotek og maler i MS-VS 2017 15.9.

### Aktuelle kapitler i boka:

- Kapittel 0, 1 og 2 i Programming – Principles and Practice Using C++ (Second Edition)

```
// Dette er et enkelt program som du kan kopiere og bruke i denne øvingen.
#include <iostream>
using namespace std;
int main() {
    cout << "Hello World!\n";
    return 0;
}
```

Figur 1: Eksempelprogram Helloworld.cpp

## Oppgave 0 - Installasjon

Vi beskriver her kort hvordan du kan laste ned **MS-VS Community Edition 2017**. Les hele avsnittet før du setter i gang. Gå til <https://visualstudio.microsoft.com/downloads/> og velg «Free Download» under «Community». Pass på at det står Visual Studio 2017 på websiden. Da lastes det ned en fil som heter `vs_community_999.exe` der 999 er ett eller annet langt nummer. Denne filen<sup>1</sup> legges normalt i en «download-folder», og du kan dobbeltklikke på den for å starte opp «Visual Studio Installer».

Etter en liten stund får du opp en meny der du blir bedt om velge hva du vil installere. Velg «Desktop development with C++». Før du trykker «installer», se i listen under «Installation Details» og sørg for at «Windows SDK (10.0.17763.0)» er markert. De andre valgene trenger du ikke for dette kurset og de vil bare bruke av plassen du har på disken i PCen din. Bare C++ alene vil kreve 6.16 GByte ledig plass. Trykk Install-knappen nede til venstre. (Ikke prøv å installere Enterprise eller Professional versjonen.) Installasjonen tar kanskje 15-20 minutter på en bra PC med et bra trådløst nettverk, men det kan ta mye lenger tid om du sitter sammen med andre på en studentsal og laster ned. Av denne grunn er det **viktig at du prøver å inninstallere på forhånd og ikke i en forelesning eller øvingstimen til studassen din**. Verktøyet er gratis, men krever at du har en Microsoft konto. Det kan du opprette gratis, og du trenger ikke bruke den til annet enn for bruk av MS-VS, dersom du ikke ønsker det.

Hvis du allerede har Visual Studio installert anbefaler vi at du oppdaterer Visual Studio med programmet Visual Studio Installer. Det gjør det enklere å følge forelesninger, der vil foreleser benytte siste versjon. Samtidig vil det gi mindre hodebry i denne øvingen.

## Oppgave 1 - Programmering med Visual Studio 2017

I denne oppgaven antas det at du allerede har installert Visual Studio 2017, som beskrevet i Oppgave 0.

### Oppgave 1.1 - Mitt første prosjekt i Visual Studio

1. Start Visual Studio 2017 fra startmenyen/startskjermen.

---

<sup>1</sup>Den 11 Desember 2018 er det versjon 15.9.4 du får installert, men det må forventes mindre oppgraderinger ganske jevnlig. Merk at **selv bare en oppdatering til nyere versjon kan kreve en større nedlasting og ta mye tid!**. (Det finnes en gratis 2019 versjon av MS-VS også, men den er ikke Community Edition og kan bli stengt eller kreve betaling snart, så den anbefales *ikke*.)

2. Velg **File** → **New** → **Project** og opprett et **Visual C++** prosjekt av typen **Empty Project**. Kall prosjektet f.eks. «HelloWorld», og lagre prosjektet i en egnet mappe.
3. I vinduet som heter **Solution Explorer** vil du se et mappetre, blant annet med en mappe kalt «Source Files». Høyreklikk på denne mappen og velg **Add** → **New Item**. Velg deretter **C++ File (.cpp)**. Kall den for eksempel `HelloWorld.cpp`, og trykk **Add**.
4. Du vil nå få opp den tomme filen du nettopp lagde. Kopier inn kildekoden fra figur 1 og lagre.
5. Kompilering i VS gjøres ved hjelp av menyvalget **Build** → **Build Solution** (eller ved å trykke **Ctrl+Shift+B** eller funksjonstast **F7**). Prøv å finne `HelloWorld.exe` (filnavnet avhenger av hva du kalte prosjektet) som nå ble opprettet (tips: let i mappen du lagret prosjektet ditt i).
6. Kjør programmet ditt ved å trykke **Ctrl+F5**, eller ved å gå inn i menyen **Debug** og velge **Start Without Debugging**. Hva skjer? Jo, programmet ditt blir kjørt, men det blir umiddelbart ferdig og avsluttes, slik at du ikke fikk se teksten du skrev ut! Dette hadde ikke vært et problem om programmet ditt forventet input fra brukeren, men her er det eneste som skjer at en tekststreng skrives til skjermen, før programmet terminerer.
7. For å fikse dette er vi nødt til å endre litt i innstillingene til prosjektet du opprettet. I mappetreet ditt i «Solution Explorer» ser du øverst en «Solution», og under det navnet på prosjektet ditt (f.eks. «HelloWorld» med fete typer), som inneholder alle mappene og filene i prosjektet. Høyreklikk på prosjektet (**HelloWorld**) og velg **Properties**. I vinduet som kommer opp går du inn i **Configuration Properties** → **Linker** → **System**. Klikk nå på linjen der det står «SubSystem». Da skal det dukke opp en liten grå firkant til høyre på samme linje. Trykk på den for å få opp en ny meny, og i den velger du (CONSOLE/SUBSYSTEM:CONSOLE). Klikk **OK** for å lagre innstillingene.
8. Kjør programmet ditt på nytt ved å trykke **Ctrl+F5**. Hvis du får opp et spørsmål om prosjektet skal bygges på nytt, trykk **Yes**. Du ser nå at programvinduet blir værende, og du kan se teksten som ble skrevet ut.

## Oppgave 1.2 - Kompilering og feilmeldinger

En god del av tiden du bruker på å gjøre øvinger vil bestå i å finne ut hva som er feil i koden din. En type feil er syntaktiske feil som resulterer i kode som ikke vil kompilere. Hvis du prøver å kompilere kode som ikke er riktig skrevet vil kompilatoren gi deg feilmelding(er) som inneholder informasjon om hva som kan være galt. Noen ganger er dette forståelig informasjon, andre ganger kan det være vanskeligere å skjønne feilmeldingene.

Du skal nå med vilje «ødelegge» koden din ved å endre på småting, og deretter observere hva slags feilmeldinger du får når du kompilerer. Du kan for eksempel gjøre følgende:

1. Fjern semikolonet på slutten av linjen som skriver ut teksten «Hello World!» og trykk **F7** for å kompilere på nytt.
2. Forstår du feilmeldingen?

3. Introduser andre feil og les feilmeldingene som kompilatoren gir. Du kan f.eks. prøve å slette en av krøllparantesene, skrive `cout` som `cut` osv.
4. Prøv å dobbeltklikke på linjer i feilmeldingen for å se hvordan Visual Studio merker linjene i koden din der den tror feilen ligger.
5. Husk å rette opp filen igjen før du går videre.

## Oppgave 1.3 - Tre ulike prosjekttyper i øvingsopplegget

I forbindelse med øvingsopplegget i emnet TDT4102 vil du ha behov for å beherske tre ulike prosjekt-typer. Vi vil nedenfor beskrive hvordan du går frem for hver av disse.

### 1. Enkelt konsoll-prosjekt

Dette er et helt enkelt prosjekt som bare bruker tekst slik som det du laget i innledingen av denne oppgaven. Det trenger ikke nærmere forklaring.

### 2. Konsoll-prosjekt med `std_lib_facilities.h`

Læreboka antar at du benytter en spesiell include-fil som heter `std_lib_facilities.h` og som er på flere hundre linjer. Hensikten med den er å gjøre programmeringen lettere for deg ved å skjule en del detaljer samt «rigge til» en pedagogisk omgivelse. Det er viktig å bruke den for å kunne følge lærebokas eksempler på en god måte, og også for å kunne kjøre kode-eksempler fra forelesningene. I første del av kurset trenger vi ikke bry oss om innholdet i denne fila, men bare forholde oss til det som fremgår av forelesningene og i læreboka.

For å gjøre det enklest mulig å opprette et prosjekt av denne typen har vi laget en såkalt prosjekt-mal (Eng. `project-template`) som setter opp et Visual Studio prosjekt slik at du har tilgang til include-filen. (Mot slutten av kurset vil vi forstå mer av innholdet i include-fila, og også hvordan vi evt. vil kunne klare oss uten den.) Gå frem på denne måten:

- (a) Last ned fila `TDT4102.zip` som du finner på Blackboard under Øving 0. Denne kreves for både denne prosjekttypen og det vi kaller PPP-FLTK-prosjekt i neste punkt.
- (b) Pakk ut zip-filen til `C:\Program Files\`  
Legg først zip-filen i `Program Files` katalogen (mappen), deretter høyreklikker du på den og velger `Extract all...` I vinduet som dukker opp må du fjerne `TDT4102` som står bak `C:\Program Files\` før du trykker på «Extract» knappen.
- (c) Sjekk at stien til `std_lib_facilities.h` er  
`C:\Program Files\TDT4102\includes\std_lib_facilities.h`.
- (d) Last ned zip-filen `TDT4102_std_lib_facilities.zip` fra Blackboard og legg den i templatemappa til Visual Studio. Templatemappa er `C:\Users\<brukernavn>\Documents\Visual Studio 2017\Templates\ProjectTemplates\Visual C++ Project\` som default. Zip-fila skal ikke pakkes ut, men legges direkte i templatemappa.
- (e) Det skal nå være mulig å velge `TDT4102_std_lib_facilities` som en prosjekttipe når man starter et nytt prosjekt i Visual Studio.

- (f) Test at alt er blitt satt opp riktig ved å lage et prosjekt som bruker denne templatene og lim inn eksempelkoden fra filen `old_hello.cpp` som du finner på Blackboard. (Vi skal i Oppgave 1.4 forklare hvordan du kan laste ned *alle eksempelprogram* som vil bli benyttet underveis i dette emnet etterhvert som de lages). Det som er nevnt i punkt 7 under oppgave 1.1 er ikke nødvendig å utføre her da «Console» er satt opp som «Sub-System» i denne prosjekt-malen. Det innebærer at kallet på `keep_window_open()` egentlig er unødvendig.
- (g) Litt avhengig av PC-en din, og når du installerte Visual Studio, kan det hende du ikke kan kjøre koden og får en lang liste feilmeldinger. For å løse dette kan du høyreklikke på prosjektet i «Solution Explorer» og velge «Retarget Projects». Trykk deretter på Ok i vinduet som kommer opp. Du skal nå kunne kjøre koden som vanlig. Merk at du må gjøre dette for alle prosjekter du lager basert på denne malen.

Prosjekter av denne typen kan ikke ha grafikk.

### 3. PPP-FLTK-prosjekt

Læreboka, som ofte forkortes til PPP etter tittelen på boka, benytter et gratis og enkelt grafikk-bibliotek som heter FLTK. FLTK står for Fast and Light Tool Kit, og biblioteket er kryss-plattform, dvs. det er tilgjengelig under Windows, Mac-Os og linux. Lærebokforfatteren har laget et lite grafikk-bibliotek oppå FLTK, og igjen er dette gjort for å forenkle. Vi ønsker å benytte dette for å følge læreboka tettest mulig. Vi har laget en MS-Visual Studio prosjektmal som dere benytter i alle øvinger som har grafikk.

Denne settes opp på samme måte som `std_lib_facilities`-prosjektet i punkt 2 ovenfor. Siden alle bibliotekfilene som kreves ble lastet ned og lagt på riktig plass når forrige template ble satt opp kan man hoppe rett til punkt (d). Altså:

- (a) Last ned zip-filen `TDT4102_Graphics.zip` og legg den i template-mappen til Visual Studio. Zip-fila skal ikke pakkes ut, men legges direkte i templatemappa.
- (b) Det skal nå være mulig å velge `TDT4102_Graphics` som en prosjekttype når man starter et nytt prosjekt i Visual Studio.
- (c) Test at alt er blitt satt opp riktig ved å lage et prosjekt som bruker denne templatene. Her er et eksempelprogram lagt inn som en del av templatene (i `main.cpp`) så du kan kjøre programmet direkte. Du skal få opp en rød trekant i et vindu med en knapp som heter «Next». Trykker du på den vil grafikk-vinduet forsvinne.
- (d) Litt avhengig av PC-en din, og når du installerte Visual Studio, kan det hende du ikke kan kjøre koden og får en lang liste feilmeldinger. For å løse dette kan du høyreklikke på prosjektet i «Solution Explorer» og velge «Retarget Projects». Trykk deretter på Ok i vinduet som kommer opp. Du skal nå kunne kjøre koden som vanlig. Merk at du må gjøre dette for alle prosjekter du lager basert på denne malen.

## Oppgave 1.4 - Eksempelprogrammer fra forelesningene

Underveis i semesteret vil vi legge ut eksempelprogrammene fra forelesningene utenfor Blackboard. Du vil finne dem ved å gå til <https://github.com/LasseNatvig/cppc> i en webleser. Her velger du f.eks. folderen `lec1` hvis du vil se eksemplene fra forelesning 1. Trykk videre på folderen med det aktuelle eksemplet og du vil få frem en eller flere filer som hører til dette eksemplet. Herfra kan du velge og kopiere ut koden, for så å lime inn i Visual Studio for å kompilere og kjøre eksemplet.

GitHub er et anerkjent system for samarbeid og deling av kode. Ved å legge eksempelprogrammene her vil studentene hele tiden ha direkte adgang til siste versjon av koden, som ofte vil kunne bli justert like før, under og etter forelesningene. Måten vi foreslår for å laste ned koden er en enkel «nybegynner-metode», siden bruk av Git og GitHub *ikke* er del av pensum i dette faget.

**Oppgave:** Gå til forelesning nr. 2 og velg mappen for eksemplet `graph_3`. Opprett ett PPP-FLTK-prosjekt og kopier innholdet i `.cpp`-fila inn i `main.cpp`. Kompiler og kjør programmet. Du kan også endre navn på fila fra `main.cpp` til f.eks. `graph_3.cpp`

## Oppgave 1.5 - Kompilering fra kommandolinje

Skriving av kildekode og kompilering kan i prinsippet gjøres med enkle verktøy. En helt vanlig teksteditor er alt du trenger for å skrive kode, og kompilering kan gjøres ved å kjøre kompilatoren fra kommandolinja.

Vi vil her anta at du bruker Microsofts kompilator `cl`, som følger med Visual Studio.

1. Start en vanlig teksteditor<sup>2</sup>, og kopier og lim inn eksemplet fra første del av øvingen. Opprett en ny katalog i hjemmemappen din og lagre fila din der, for eksempel med navnet `HelloWorld.cpp`. Sjekk mappen hvor fila din er lagret, og forsikre deg om at den er der og har riktig navn.

Det er vanlig konvensjon at kildekodefiler for C++ har filtypenavnet (delen av filnavnet etter punktum) `cpp`. Ved å bruke riktig filtypenavn oppnår du at mange verktøy automatisk skjønner at fila inneholder C++-kode. Hvis du bruker et annet filtypenavn kan det hende at du ikke får compilert koden.

2. For å compilere fra kommandolinjen trenger vi først å starte opp et kommandolinjevindu. Dersom du åpner kommandolinjen på vanlig måte med `cmd.exe`, vil man hver gang man ønsker å kjøre C++-kompilatoren `cl` måtte spesifisere hvilken mappe denne ligger i. Det er tungvindt. I stedet skal vi bruke «Developer Command Prompt for VS2017». For å starte denne, gå inn på start-menyen (Windows 7/10) eller start-skjermen (Windows 8) og begynn å skrive «Developer Command Prompt». Du skal få opp programmet i listen – start dette.
3. Vi skal nå navigere oss fram til mappen der du lagret `HelloWorld.cpp`. For å navigere mellom mapper i Windows-kommandolinjen bruker man kommandoen `cd` (change directory).
4. Når du er kommet til mappen der `HelloWorld.cpp` ligger, skal vi compilere denne til en programfil. For å gjøre dette skriver du

```
cl HelloWorld.cpp
```

5. Sjekk nå hva som ligger i mappen og se hvilke filer som er produsert. `obj`-fila er en midlertidig fil du ikke trenger å bry deg om, mens `exe`-fila er det endelige programmet ditt. Kjør programmet du har laget ved å skrive `HelloWorld.exe`, og sjekk utskriften.

---

<sup>2</sup>Ett forslag kan være den du finner på <https://notepad-plus-plus.org/>