



Norges teknisk–naturvitenskapelige
universitet
Institutt for datateknologi og
informatikk

TDT4102 Prosedyre- og objektorientert programmering Vår 2019

Øving 0 for Mac

Frist: Ingen (frivillig øving)

Mål for denne øvingen:

- Bli kjent med programmeringsverktøy
- Lage et første program med Xcode
- Kunne laste ned og kjøre eksempelprogram fra forelesningene med Xcode
- 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, compilere, 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.**

NB: Denne øvingen er testet for Xcode versjon 10.1 på en ganske ny Mac med mac OS High Sierra 10.13.6 og en med macOS Mojave versjon 10.14.2. Dersom du bruker eldre versjoner av Xcode kan det være at ting er annerledes, og står du fast bør du poste en melding på diskusjonsforum i Blackboard. Vi vil prøve å hjelpe med andre versjoner, men anbefaler sterkt at du oppgraderer til nyeste versjon både av Xcode og macOS. Du kan sjekke hvilken versjon av Xcode du kjører ved å velge **Xcode** → **About Xcode** i menylinjen.

Aktuelle kapitler i boka:

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

Oppgave 0 - Installasjon

Vi vil våren 2019 benytte verktøyet Xcode 10 som er gratis. Dersom du bruker en tidligere versjon av programmet er det sannsynlig at deler av den følgende beskrivelsen ikke stemmer. Start med å installere Xcode. Dette kan gjøres fra Mac App Store. Dersom du installerte Xcode for en stund siden, anbefales det sterkt at du oppgraderer til siste versjon av programvaren.

Oppgradering av operativsystem eller Xcode, eller installasjon av Xcode kan innebære tidkrevende nedlasting. Av den grunn er det **viktig at du prøver å gjøre dette på forhånd, og ikke i en forelesning eller øvingstimen til studassen din.**

Oppgave 1 - Programmering med Xcode

I denne oppgaven antar vi at du allerede har installert Xcode, som beskrevet i Oppgave 0.

Oppgave 1.1 - Mitt første prosjekt i Xcode

1. Start Xcode, ikonet til programmet er lett å kjenne igjen med bildet av en hammer.
2. Velg «Create a new XCode project».
3. Under «macOS» (evt. «OS X») og «Application» velg «Command Line Tool».
4. Veiviseren ber deg nå fylle ut en rekke felter. Her skal du skrive inn et produktnavn, f.eks. «Oving0» eller «HelloWorld». Organisasjonsnavn og organisasjonsidentifikator skal også fylles ut – her betyr det ikke noe hva du skriver. Du trenger ikke velge noe «team account». Pass på å velge **C++** fra språkmenyen. Lagre prosjektet i en egnet mappe. Du kan fjerne avkrysningen for «Create Git Repository» dersom du ikke vet hva dette er, eller ikke ønsker å bruke verktøyet.
5. Xcode har nå opprettet et prosjekt og en standard kildekodefil kalt `main.cpp`. Du får frem den C++ filen i editoren ved å dobbelt-klikke på `main.cpp` i filtreet til venstre i applikasjonen.
6. I sidelinjen har du nå et prosjekt med navnet du valgte i veiviseren. Dette prosjektet har to mapper under seg: en med navnet «Products», og en med samme navn som prosjektet.
7. Kompiler programmet ditt uten å kjøre det ved å trykke **Product** → **Build** i menylinjen (hurtigtast: **CMD+B**).
8. Kompilering etterfulgt av kjøring av programmet i Xcode gjøres ved å trykke på «play»-knappen) øverst til venstre i Xcode (hurtigtast: **CMD+R**). Gjør dette nå.
9. Du ser at programmet kjører og skriver ut teksten «Hello World!» i bunnen av programvinduet. Gratulerer!
10. Du kan gjerne redigere programmet slik at det blir helt likt det i Figur 1, det gir samme resultat.

```
// 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

Når du kompilerer programmer i Xcode vil det lages en programfil, som havner på et litt annet sted enn prosjektet ditt. For å finne denne, åpner du mappen «Products» i sidelinjen. Inni denne mappen ligger det et objekt med samme navn som prosjektet ditt og med et svart «terminalikon». Høyreklikk på dette objektet og velg **Show in Finder** for å finne ut hvor det ligger.

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 kompiler 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 innledningen 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 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 for Mac**. Denne kreves for både denne prosjekttypen og det vi kaller PPP-FLTK-prosjekt i neste punkt.
- (b) Pakk ut zip-filen på et sted du velger selv, f.eks. på desktop. Merk at dersom du bruker Safari pakkes filen ut av seg selv. Kopier deretter katalogen som heter `tdt4102` inn i katalogen som heter `/Library` (Evt. `/Bibliotek` om du har norsk versjon). Sørg for at `tdt4102`-mappen har nøyaktig dette navnet, slik at den fullstendige stien blir `/Library/tdt4102`. En måte for å finne `/Library` er å åpne Finder, velge «Gå» oppe i menylinja, velge «Gå til mappe», og så skrive inn `/Bibliotek` (snarvei: `cmd+shift+g` og lime inn «`/Library`» - inkludert skråstrek).
Det er en vanlig feil å legge katalogen i `/Users/<brukernavn>/Library`. For at prosjektet skal fungere *må* katalogen `tdt4102` ligge som beskrevet over. Får du feil om at prosjektet ikke finner f.eks. `std_lib_facilities.h`, er feil plassering av denne mappen årsaken.
- (c) Last ned zip-filen `std_lib_facilities_mac.zip` fra Blackboard og legg den der du ønsker. Hver gang du skal lage et nytt prosjekt av denne typen pakker du opp ZIP fila ved å dobbeltklikke på den. Gå inn i folderen som opprettes, og dobbeltklikk på prosjekt-fila.
- (d) Prosjektet har en fil som heter `main.cpp` som er omtrent helt tom.
- (e) Test at alt er blitt satt opp riktig ved å lime 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).
- (f) Får du feilmelding om at prosjektet ikke finner f.eks. filen `std_lib_facilities.h`, vær sikker på at punkt 2b) er utført som beskrevet.

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 tetttest mulig. Vi har laget en 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 (c). Altså:

- (a) Last ned zip-filen `FLTK-template.zip` og legg den der du ønsker å ha programmene dine.
- (b) Hver gang du skal lage et nytt prosjekt av denne typen pakker du opp ZIP fila ved å dobbeltklikke på den. Gå inn i folderen som opprettes, og dobbeltklikk på prosjekt-fila.
- (c) Prosjektet inneholder et grafikk-eksempel fra forelesning som kompilerer og kjører¹
- (d) Denne prosjektmalen skal brukes på alle øvinger som har grafikk, prøv gjerne å bytte ut `main.cpp` med et av eksempelprogrammene fra forelesning som har grafikk
- (e) Får du feilmelding om at prosjektet ikke finner f.eks. fila `std_lib_facilities.h`, vær sikker på at punkt 2b) er utført som beskrevet.

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 et prosjekt 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_2`. 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_2.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.

1. Start en vanlig teksteditor. TextEdit er en veldig enkel teksteditor som følger med Mac og kan brukes her. For å skrive kode i TextEdit må man, før man begynner å skrive kode, velge **Format** → **Konverter til ren tekst**, slik at det ikke lagres ekstra informasjon om

¹I skrivende stund får vi 54 warnings (iusses) på prosjektet, men prosjektet kompiles og kan kjøres. Dette er foreløpig som forventet.

stil og format i kodenfilen. Kopier og lim inn eksemplet fra oppgave 1.1. Opprett en ny mappe og lagre filen din der, for eksempel med navnet `HelloWorld.cpp`. Sjekk mappen hvor filen din er lagret, og forsikre deg om at den er der og har riktig navn.

2. Start opp et terminalvindu. Terminalen finner du i **Finder** → **Programmer** (i venstre sidemeny) → **Verktøy** → **Terminal**. Alternativt kan du også søke etter **Terminal** i Launchpad eller Spotlight.
3. Terminalvinduet starter i hjemmemappen din. Dersom du ikke lagret `HelloWorld.cpp` direkte i hjemmemappen din, må du flytte deg til riktig mappe. Dette gjør du med kommandoen `cd` (change **d**irectory). Hvis du for eksempel la filen i «Dokumenter»-mappen din, vil du måtte skrive:

```
cd Documents
```

Du befinner deg nå i «Documents»-mappen din. (Merk at dersom du kjører norsk utgave av MacOS vil denne, og tilsvarende mapper som «Pictures», alltid ha engelsk navn, uavhengig av hva den heter når du ser den i Finder.)

4. Skulle filen befinne seg i en undermappe, gjentar du `cd`-kommandoen, denne gangen med navnet til undermappen, for å gå videre dit. Ønsker du å sjekke hvilken mappe du befinner deg i, kan du gjøre dette med kommandoen `pwd`.

Dersom du skriver kommandoen `cd` uten noe etterpå (det vil si uten noen *argumenter*) vil du bli returnert til hjemmemappen din. Ønsker du å gå til mappen *over* den du befinner deg i, kan du skrive `cd ..` (to punktum). «Over» refererer her til over i *mappehierarkiet*. Befinner du deg i mappen `/Users/dittbrukernavn/Documents/tdt4102`, vil mappen over være `/Users/dittbrukernavn/Documents`.

5. Skriv kommandoen `ls` (**l**ist) for å se hvilke filer (eller mapper) som ligger i mappen du befinner deg i.
6. Vi skal nå sjekke at «Command Line Tools» er installert. Dette skal i utgangspunktet følge med nyere versjoner av XCode. For å sjekke at alt er installert kan du skrive `clang++` i terminalvinduet. Om «Command Line Tools» *ikke* er installert, får du en feilmelding som sier `Command not found`, samtidig som du får en forespørsel om du vil installere verktøyet. Installer verktøyet. Dersom du får (feil-)meldingen `clang: error: no input files` er alt installert.
7. Har du funnet fram til mappen der `HelloWorld.cpp` ligger, er du klar til å kompilere programmet. Kompilatoren som følger med Xcode heter `clang++`, og du bruker denne til å kompilere programmet ditt ved å skrive:

```
clang++ HelloWorld.cpp
```

8. Sjekk nå innholdet av mappen med `ls`, og se hvilke filer som ble produsert da du kompilerte. `a.out`-filen er programmet ditt. Kjør programmet du har laget ved å skrive `./a.out`. Hvis alt har gått bra skal utskriften fra programmet ditt vises i terminalen.
9. Rediger teksten i `HelloWorld.cpp` slik at programmet skriver ut noe annet (husk å ha med hermetegnene rundt teksten).
10. Kompiler og kjør på nytt.