Norges teknisk–naturvitenskapelige universitet
Institutt for datateknologi og informatikk

TDT4102 Prosedyre- og objektorientert programmering Vår 2020

Øving 0 for Windows

Frist: som for Øving 1

Mål for denne øvingen:

- Bli kjent med programmeringsverktøy
- Lage et første program med Visual Studio Code (VS **Code**)
- Kunne laste ned og kjøre eksempelprogram fra forelesningene med VS **Code**
- 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 2020 benytte verktøyet VS **Code** som er gratis og som kan brukes under Windows, MacOS og Linux. Forelesningene vil da bedre kunne dekke det verktøyet alle studentene bruker.

Aktuelle kapitler i boka:

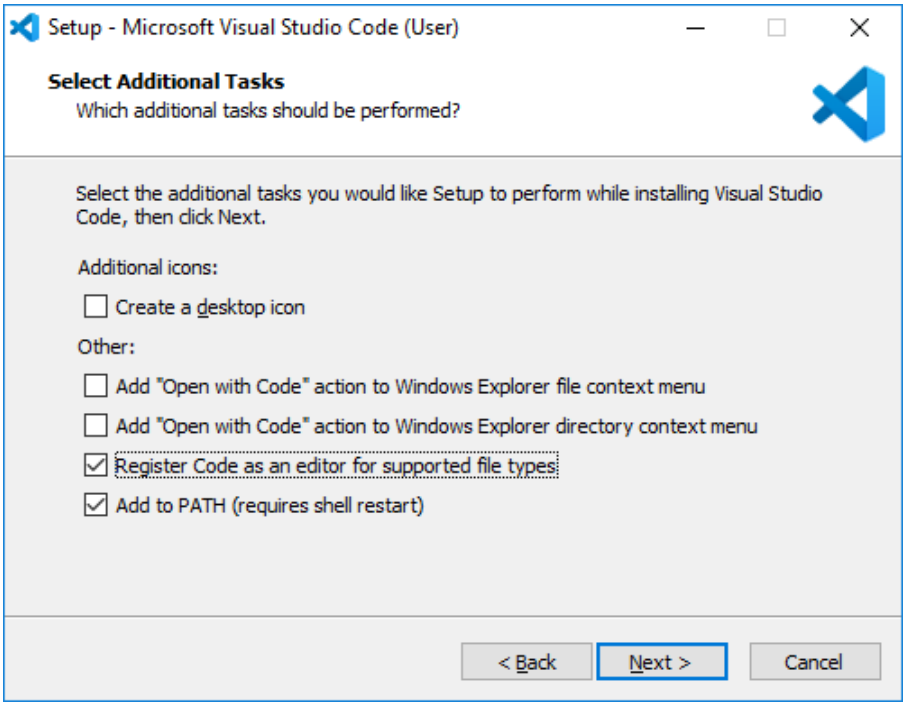
- Kapittel 0, 1 og 2 i Programming – Principles and Practice Using C++ (Second Edition)

Oppgave 0 - Installasjon

Les gjennom hele installasjonsinstruksen før du begynner.

Installasjonen krever at du installerer to programmer, VS **Code** (**Visual Studio Code**) for å skrive og redigere kode, og **clang** for å kompilere å kjøre koden. I tillegg trengs det noen filer for å kunne bruke kode fra pensumboka. Disse kan du finne i TDT4102-mappen på blackboard.

VS **Code** er en teksteditor laget for å skrive og redigere kode i forskjellige programmeringsspråk. Den kan lastes ned og installeres fra https://code.visualstudio.com. Her er det bare å trykke den store blå knappen med teksten download for windows og kjøre installeren. Man må godta en avtale fra Microsoft. Videre får man valget om hvor man vil installere programmet, her er defaultforslaget bra. I neste meny må man huke av for å *ikke* lage en mappe i startmenyen. Vi kommer til å bruke en egen snarvei for å åpne Code i løpet av øvingsopplegget. I det siste steget er det viktigste at det er en hake på "Add to PATH" og **"Register code as an editor for supported file types"**. "Add Code to PATH" lar code åpnes fra terminalen (dette er nødvendig for noe av oppsettet i faget). "Register code as an editor for supported file types" gjør at windows bruker Code for å åpne en del filtyper automatisk. Se figur 1. Trykk install og vent til den er ferdig. Fjern haken på "Launch Visual Studio Code" før du avslutter.



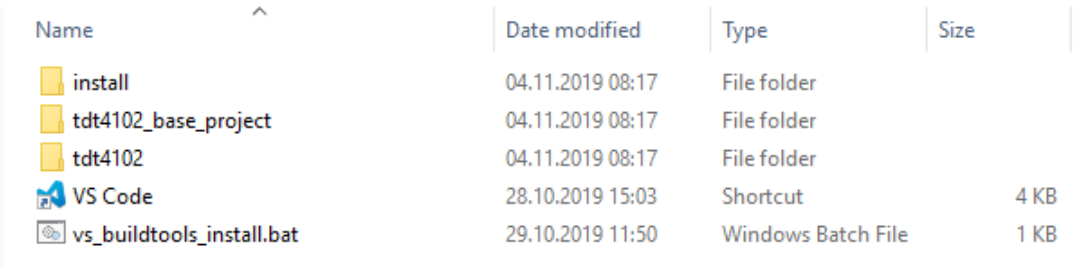
Figur 1: VSCode installasjonsoppsett

Last deretter ned filen `TDT4102_Windows.zip` fra Øving 0 i Blackboard og pakk den ut et sted du husker, for eksempel kan du lage en mappe for faget under dokumenter. Inni `TDT4102_Windows`-mappen ligger det tre mapper og to filer (se figur 2). Undermappen med navn tdt4102 skal flyttes til `C:\Program Files`. Pass på at den blir riktig plassert, da det er veldig viktig for at installasjonen skal fungere. Full sti til mappen må altså være `C:\Program Files\TDT4102`.

clang er en kompilator, altså programmet som oversetter koden du skriver i C++ til forståelige instruksjoner for datamaskinen. For å installere clang og sette det opp til bruk på windows bruker vi filen i `TDT4102_Windows`-mappen ved navn `vs_buildtools_install.bat`. Dobbeltklikk på den for å kjøre den, da vil man nok få en advarsel om ukjent utgiver. For å gå videre

må man trykke på ”mer” og ”kjør likevel”. Det eneste programmet gjør er å bruke Microsoft sin Visual Studio Installer for å installere clang til en bestemt plass i filsystemet. Man må også godkjenne at installeren gjør endringer på maskinen før installasjonen kan starte, deretter er det bare å vente til denne blir ferdig. Dette kan ta litt tid, så bare la den stå og gå litt. installeren lukkes når alt er ferdig.

Ellers i TDT4102mappen du lastet ned fra Blackboard ligger en snarvei med navnet `VS code` for å åpne VS **Code**. Det er veldig viktig å åpne VS **Code** med denne snarveien, hvis ikke vil ikke alle programmene våre fungere, så det anbefales sterkt å flytte denne til skrivebordet eller taskbaren hvor man har den lett tilgjengelig. Videre, høyreklikk på snarveien og velg ”fest til startmeny” (”pin to start”), så kan man åpne Code derifra også. Det ligger også to mapper ved navn `install` og `TDT4102_base_project`. Disse inneholder henholdsvis installasjonsfiler og et eksempelprosjekt, og det er viktig at de ikke endres.



Figur 2: Filene i TDT4102 mappen

Oppgave 1 - Programmering med VS Code

I denne oppgaven antas det at du allerede har installert VS **Code**, som beskrevet i Oppgave 0.

Oppgave 1.1 - Mitt første prosjekt i VS Code

1. Start VS **Code** fra snarveien i TDT4102 mappen, der du pakket den ut. Her også kan man få en advarsel man må godta, det kan være greit å fjerne avhuking for å alltid vise advarselen for denne snarveien, slik at man sparer noen knappetrykk i fremtiden.

2. Det vil også åpnes et konsollvindu, dette er det bare å stenge (X) etter at VS **Code** er åpent.

3. Velg **File** → **Open Folder** og åpne mappen **tdt4102_base_project**, som også ligger der du pakket ut TDT4102.

4. Når du har åpnet mappen (trykket "select Folder") skal det dukke opp en popup nede til høyre med "recommended extensions". Trykk "install all" så vil VS **Code** installere noen utvidelser som hjelper med skriving og debugging av C++-kode. Hvis du får tilbud om "Insiders version" så velg "Don't Show Again".

5. Du kan sjekke at du har de nødvendige utvidelsene med å trykke på "extensions" ikonet i menybaren helt til venstre på skjermen. (Det nederste av fem ikoner, ser ut som fire byggeklosser.) Her bør du se "C/C++" og "Project Templates" under enabled (figur 3), eller så er de "enabled globally". Hvis de ikke er der kan du søke dem opp og installere dem selv.

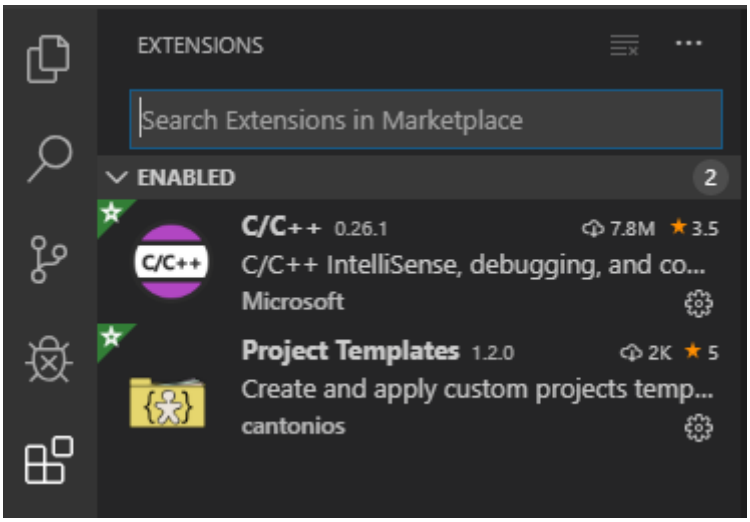
6. VS **Code** har mange tusen forskjellige utvidelser, i dette faget trenger vi bare de to som er nevnt. Det kan fort bli forvirrende med undøvendig mange, så vi anbefaler ikke å installere flere hvis du ikke vet hva du gjør. Hvis det står flere anbefalte utvidelser enn "Project Templates" og 'C/C++" er det bare å ignorere disse. Trykk på det øverste ikonet i menybaren for å komme tilbake til filutforskeren når du er ferdig.

7. Til venstre i VS **Code** vil du nå se en oversikt over filene i prosjektmappen din. Her skal det ligge en fil ved navn main.cpp, en ved navn Makefile og en mappe ved navn ".vscode".

8. Makefile og .vscode inneholder innstillinger for hvordan VS **Code** skal kompilere og kjøre koden i prosjektet ditt, og å endre dem kan gjøre at ting ikke lenger fungere som de skal. La dem derfor være som de er! Mappene "Release" og "Debug" er der Code vil legge programmene du lager etter de er kompilert, og de vil opprettes ved behov.

9. Det viktigste for oss er `main.cpp`, i denne ligger kildekoden til et enkelt program som skriver "Hello, World!" til skjermen. Hvis du trykker på main.cpp vil filen åpnes så du kan redigere den, og den vil dukke opp i listen over "open editors" øverst til venstre.

10. Kompilering i VS **Code** gjøres ved hjelp av hurtigtasten `Ctrl+Shift+B` eller **Terminal** → **Run Build Task**. Da får du opp en meny med muligheter for å bygge i debug eller release konfigurasjon. Dette kommer vi tilbake til, men nå kan du velge "Build Release Executable".



Figur 3: Utvidelsene vi trenger i faget

11. Kjør programmet ditt ved å trykke `Ctrl+F5`, eller ved å gå inn i menyen **Debug** og velge **Start Without Debugging**. Det vil åpnes et terminalvindu med teksten "Hello, World!" som blir værende til du X-er det bort, eller skriver inn en bokstav og trykker enter.

Oppgave 1.2 - Nye prosjekter i øvingsopplegget

I løpet av øvingsopplegget kommer du til å skrive mye kode i forskjellige filer. For å slippe å kopiere alle de nødvendige filene hver gang du skal lage en ny mappe vil vi bruke en utvidelse til VS **Code** som heter "Project Templates" som lar oss lage prosjekter fra såkalte maler.

For å lagre en mal som kan gjenbrukes må man:

1. Åpne DT4102_base_project i VS **Code**
2. Trykk `Ctrl+Shift+P`, du vil da få opp et vindu hvor du kan skrive kommandoer.
3. Start å skrive "Project: Save Project as Template" til du får opp det som et alternativ.
4. Trykk enter og velg hva du vil kalle prosjekttypen din. (default-navnet TDT4102_base_project er også helt greit)

Det neste du må gjøre er å lage en mappe der du ønsker å jobbe med øvingene i løpet av semesteret (for eksempel `Dokumenter/cpp/`), og deretter en "Øving 0"-mappe inne i den. Når det er gjort åpner du mappen i VS **Code** på samme måte som vi gjorde tidligere, ved hjelp av **File** → **Open Folder** og så finne fram til Øving 0 mappen du nettopp lagde. Når du har åpnet den kan du igjen trykke `Ctrl+Shift+P` og begynne å skrive "Project: Create Project from Template". Trykk enter og velg malen du har lagd, da vil mappen Øving 0 fylles med alt du trenger for å kompilere koden, og en main.cpp-fil hvor du kan programmere.

Altså, for å lage et nytt prosjekt må man:

1. Lag en ny tom mappe
2. Åpne denne mappen i Code
3. Trykk `Ctrl+Shift+P` og skriv "Project: Create Project from Template"
4. Velg TDT4102_base_project eller det du kalte prosjekt-malen.

For å teste at alt fungerer kan du kopiere koden fra figur 4 inn i `main.cpp` (Skriv over det som ligger der fra før) og teste om du får til å kompilere og kjøre. Hvis alt har gått som det skal vil det nå åpnes et nytt vindu med en rød trekant. Dette er et eksempel på enkel grafikk, som vi kommer til å bruke en del senere i øvingsopplegget.

```
// Example program from PPP, page 415
#include "Graph.h"
#include "Simple_window.h"
int main() {
        using namespace Graph_lib;
        cout << "The New \"Hello, Graphical World!\" message\n";
        Point tl{ 100, 100 };
        Simple_window win{ tl, 600, 400, "Canvas" };

        Polygon poly;
        poly.add(Point{ 300, 200 });
        poly.add(Point{ 350, 100 });
        poly.add(Point{ 400, 200 });
        poly.set_color(Color::red);

        win.attach(poly);
        win.wait_for_button();
}
```

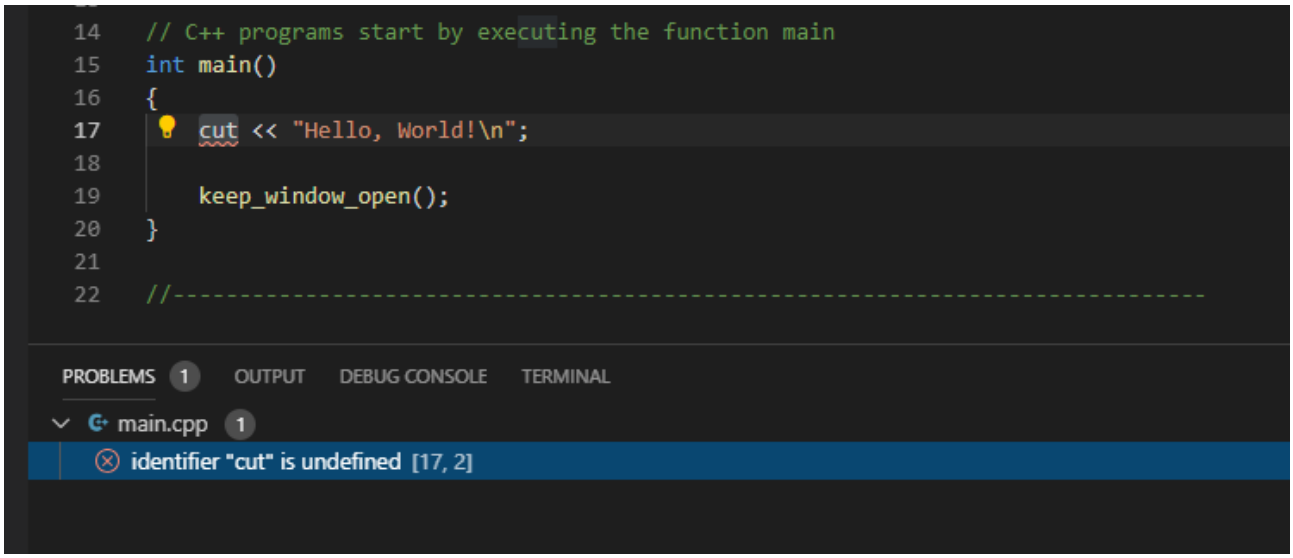
Figur 4: Eksempelprogram med enkel grafikk

Oppgave 1.3 - 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 (enkle skrivefeil) 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 vanskelig 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. Det vil nå dukke opp en feilmelding i terminalvinduet nederst, 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. Det skal også dukke opp feilmeldinger i Problems fanen på terminalen (se figur 5). Her kan du klikke på linjer i feilmeldingen for å se hvordan VS **Code** markerer linjene i koden din der den tror feilen ligger.
5. Husk å rette opp feilene i filen igjen før du går videre.



Figur 5: Problems fanen i VSCode

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/cppc2020` i en webleser. Her vil alle eksempler bli lag ut i mapper med beskrivende navn. Finn fram til eksempelet du ønsker å teste, og trykk på det så får du se kildekoden. 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: Finn eksempelet som heter "`graph_2/main.cpp`". Opprett ett nytt prosjekt, kall det for eksempel "Forelesningseksempel" (se oppskriften i slutten av Oppgave 1.3 hvis du trenger en oppfriskning) og kopier innholdet i `graph_2/main.cpp`-fila inn i din nye `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. Vi vil heller ikke benytte *Graph.h* eller *Simple_windows.h*, ettersom det er mer avansert å kompilere "for hånd". Eksempelprogrammet vi skal bruke i denne oppgaven vises i Figur 6.

Vi vil her bruke Microsofts kompilator `cl`, som fulgte med under installasjonsprossessen.

```cpp
// C++ Hello World without std_lib_facilities
#include <iostream>

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

Figur 6: Eksempelprogram uten avhengigheter på Graph.h eller Simple_window.h

1. Opprett en ny mappe i hjemmemappen din, og åpne denne i Visual Studio Code. Kopier og lim inn Hello World-eksemplet i Figur 6 i en ny fil, 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.

   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 filen inneholder C++-kode. Hvis du bruker et annet filtypenavn kan det hende at du ikke får kompilert koden.

2. For å kompilere 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 vi disse kommandoene

   - **cd** (**c**hange **d**irectory)
     - For å gå inn i en mappe skriver man: `cd <navn på mappen din>`
     - for å gå opp et nivå skriver man: `cd..`
   - **dir**
     - for å se hva som ligger i mappen man er i skriver man `dir`

4. Når du er kommet til mappen der `HelloWorld.cpp` ligger, skal vi kompilere 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`-filen er en midlertidig fil du ikke trenger å bry deg om, mens `exe`-filen er det endelige programmet ditt. Kjør programmet du har laget ved å skrive `HelloWorld.exe`, og sjekk utskriften.