

Norges teknisk–naturvitenskapelige
universitet
Institutt for datateknologi og
informatikk

TDT4102 Prosedyre- og objektorientert programmering Vår 2021

Øving 0 for Mac

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, 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.**

Vi vil våren 2021 benytte verktøyet VS Code som er gratis og kan brukes under Windows, MacOS og GNU/Linux.

Vær obs på at installering av alle programmene kan ta opptil flere timer, avhengig av hvor rask internettilkobling og PC du har.

Aktuelle kapitler i boka:

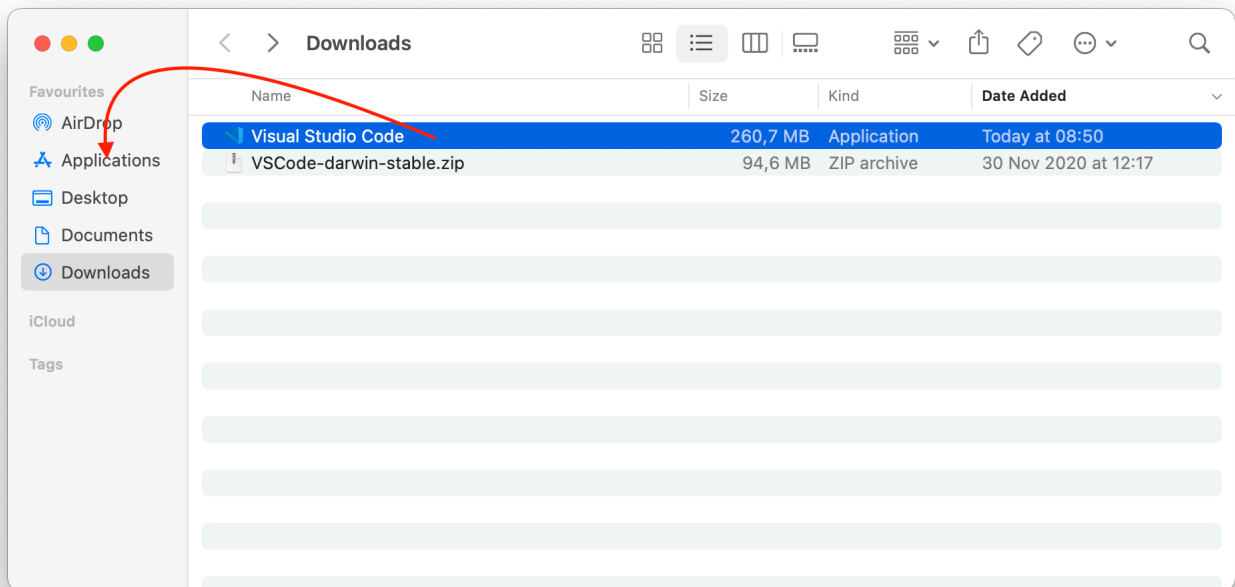
- Kapittel 0, 1 og 2 i Programming – Principles and Practice Using C++ (Second Edition)

0 Oppgave 0 - Installasjon

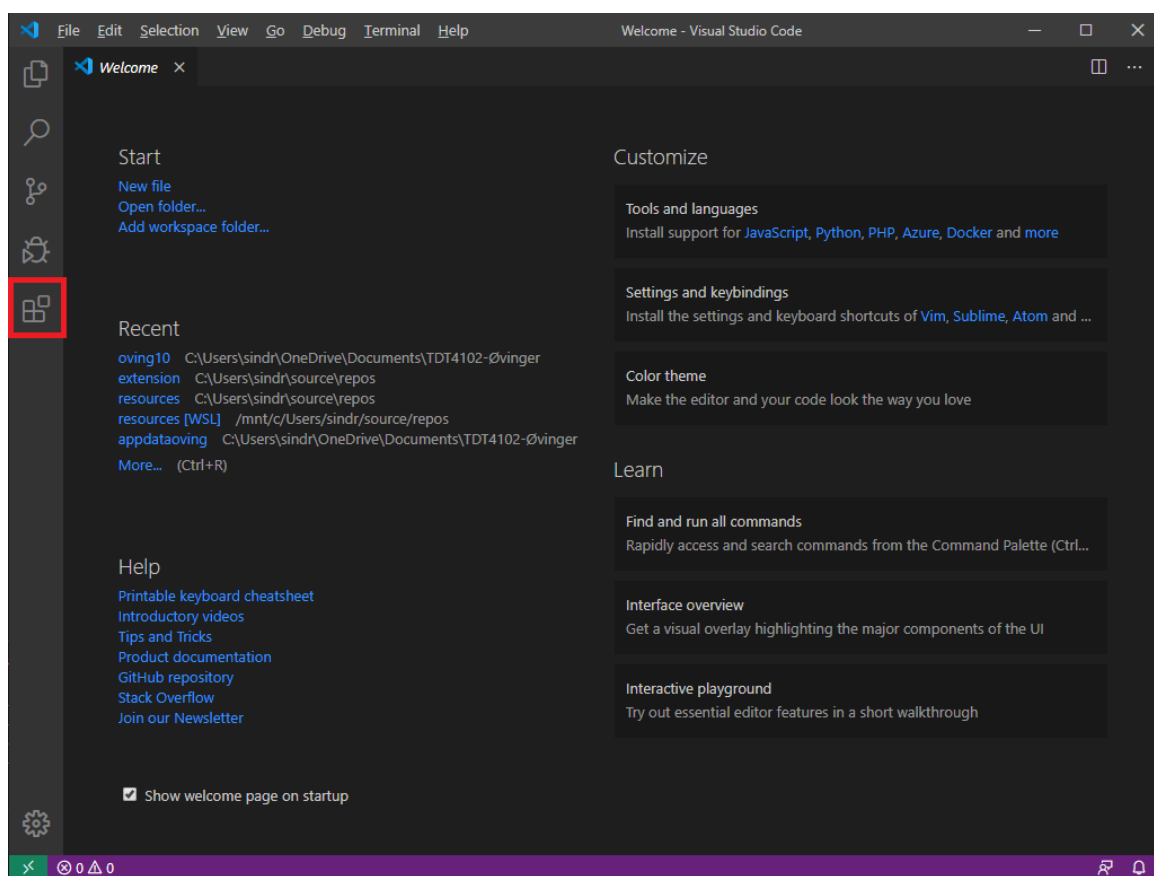
Les gjennom hele installasjonsinstruksen før du begynner.

Installasjonen krever at du installerer VS Code for å skrive og redigere kode. Oppstår det problemer underveis har vi listet opp noen kjente problemer i [Avsnitt 2](#).

VS Code er en teksteditor laget for å skrive og redigere kode i forskjellige programmeringsspråk. Programmet kan lastes ned fra <https://code.visualstudio.com/docs/?dv=osx>. Det som lastes ned er en .zip-fil som inneholder programmet. For å lettere finne programmet bør det installeres ved at du flytter programmet (filen som heter **Visual Studio Code** fra *Downloads* til *Applications*. Dette kan du gjøre i *Finder* ved å dra og slippe programmet, som vist i [Figur 1](#) (som når du installerer programmer til vanlig).



Figur 1: Installer VS Code ved å flytte programmet til **Applications**.



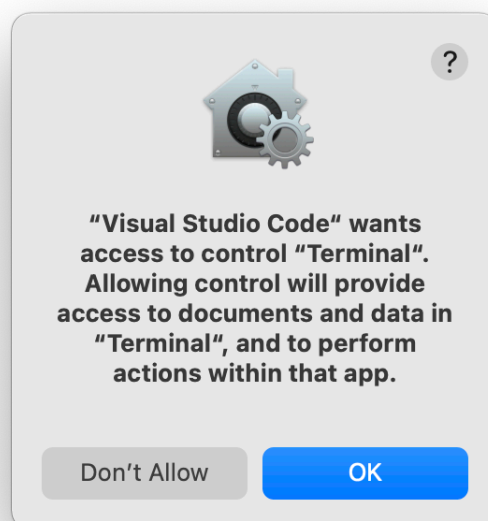
Figur 2: VS Code Extension-meny i rødt.

0.1 Installere TDT4102 Tools

VS Code vil la oss skrive og redigere kode, men vi trenger også en kompilator og bibliotek som følger med boken for å bygge og kjøre øvingene. For å forenkle denne prosessen har vi laget et tillegg (extension) til VS Code. Dette tillegget vil laste ned og starte installasjonen av de resterende verktøyene vi trenger, samt gjøre det lett å lage nye prosjekter. **Ikke lukk datamaskinen underveis i installasjonsprosessen, dette kan ta alt fra 10 minutter til et par timer.** *Det kan hende du må skrive inn passordet ditt flere ganger for at installasjonen skal kunne fullføre.*

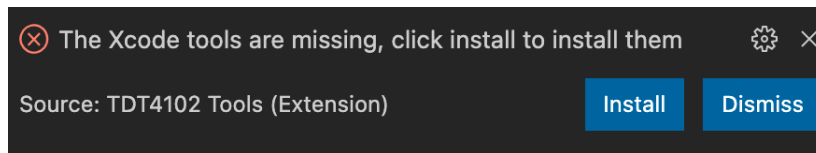
Merk at stegene i denne prosessen kan variere hvis du allerede har installert VS Code, Xcode eller andre verktøy tidligere.

1. Åpne VS Code og velg Extensions fra menyen til venstre, se **Figur 2**. (Cmd+Shift+X)
Det kan være du må trykke *Open* når du åpner programmet første gang siden programmet er lastet ned via en nettleser og ikke gjennom Apple sin *App Store*. Søk etter TDT4102 Tools, og trykk på **Install**.
Nå vil alle nødvendige programmer lastes ned og installeres. Nederst til høyre i VS Code-vinduet vil det dukke opp en noen meldinger som forteller om installeringen er underveis, er fullført eller har feilet.
2. På et tidspunkt vil du få spørsmål om å godkjenne tilgang til Terminal-applikasjonen, som vist i **Figur 3**. Godkjenn tilgangen, som lar VS Code starte programmene du har laget.

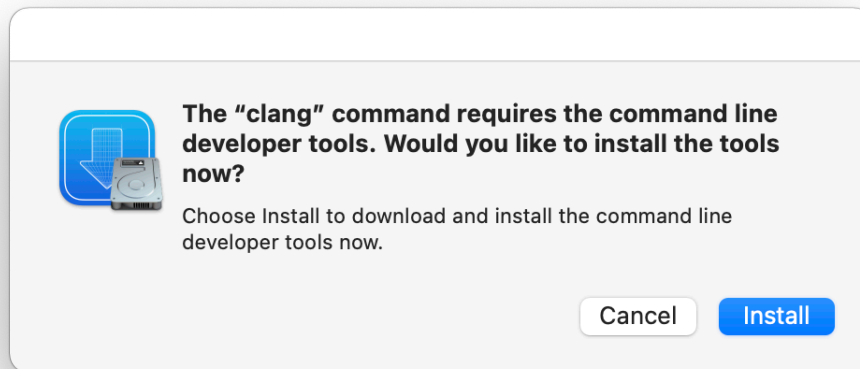


Figur 3: Trykk OK for å godkjenne VS Code sin tilgang til Terminal-applikasjon

3. I de fleste tilfeller får man en eller flere meldinger fra VS Code om at enkelte programmer eller tjenester mangler, som vist i **Figur 4** og **Figur 5**. Dersom dette skjer trykker du **Install** og deretter **Allow** eller **Install**, hvis du får spørsmål om å godkjenne tilgang.



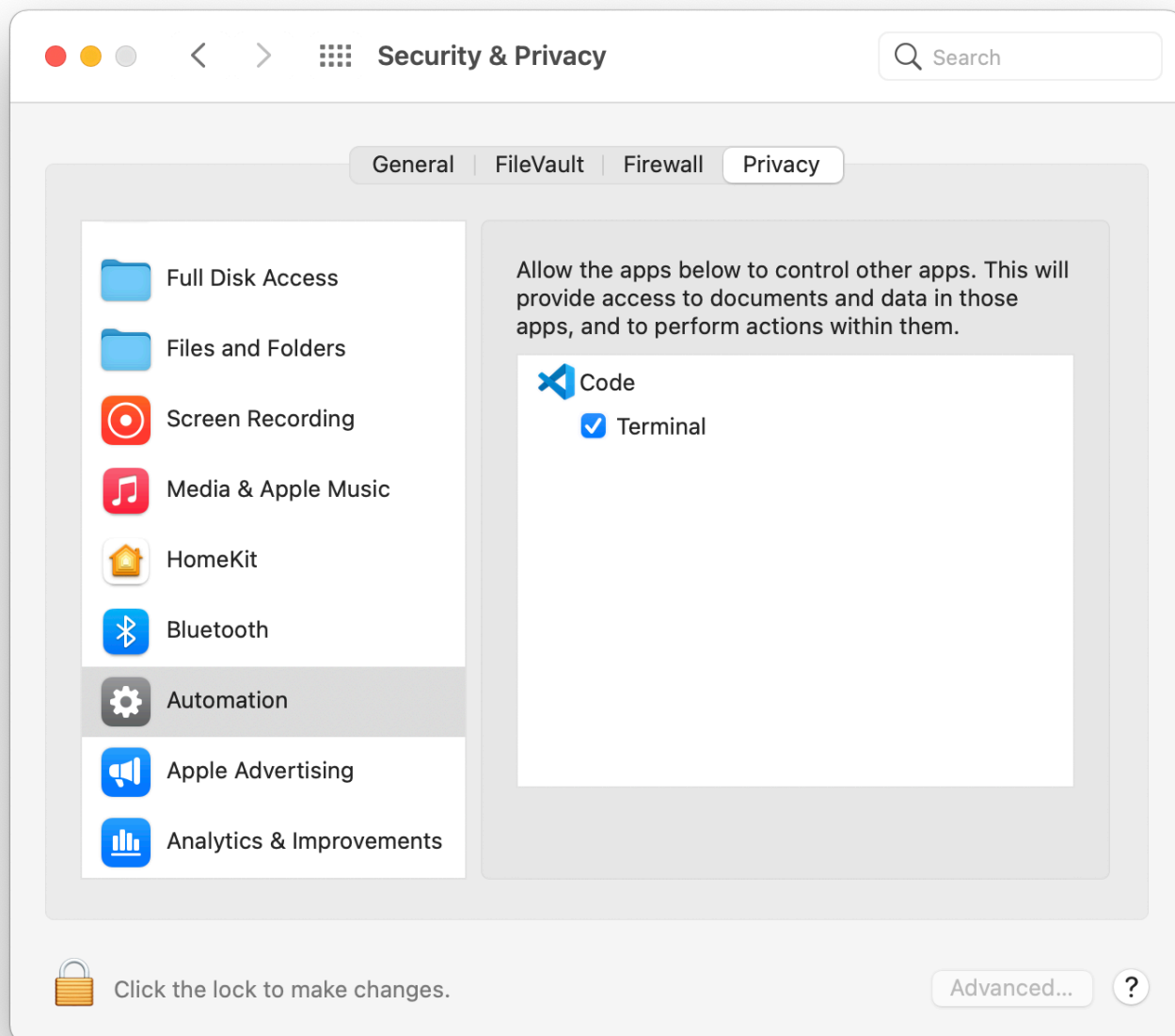
Figur 4: Manglende programmer under installasjon av extension i VS Code. Trykk **Install** og hvis du får spørsmål om å godkjenne tilgang, trykk **Allow**.



Figur 5: Trykk **Install** for å installere utviklervertøy.

4. Dersom du måtte installere Xcode-verktøy får du nå spørsmål om å godkjenne en lisensavtale, trykk **Agree** for å fortsette. Trykk **Done** når installeringen er fullført.
5. Når du har godkjent tilgangen vil det åpnet et vindu som viser at tilgangene er satt opp korrekt. Dobbelsjekk at vinduet som dukker opp samsvarer med **Figur 6**.
6. Lukk alle vinduer av VS Code før du går videre.

TDT4102-tillegget vil sjekke at du har de nyeste filene og malene, samt at alt du trenger er installert hver gang du åpner VS Code. Dersom det for eksempel skulle komme endringer i malene vil disse bli lastet ned automatisk, og du vil få en melding nederst i høyre hjørne av VS Code om at malene er oppdatert.



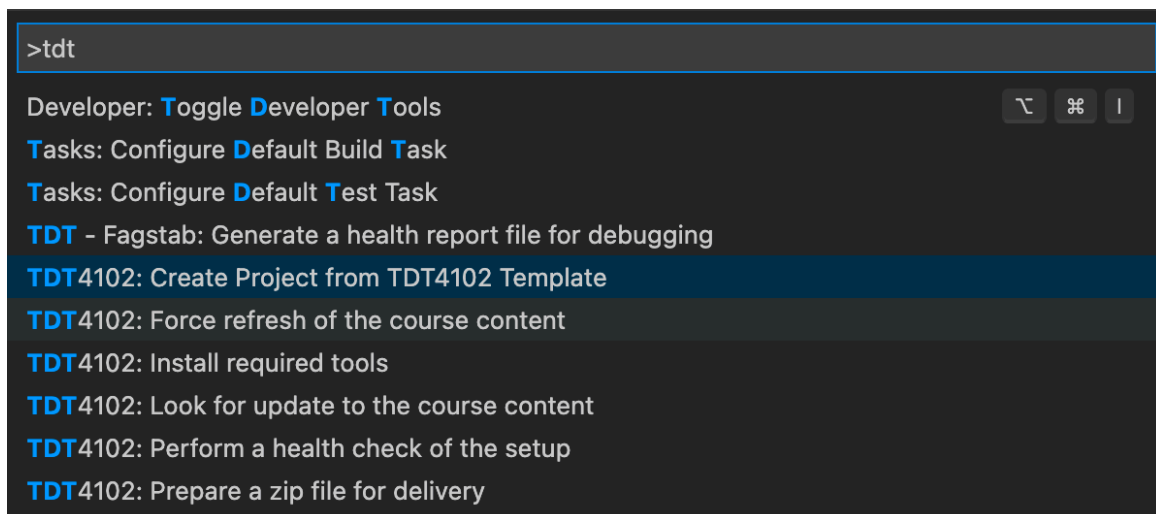
Figur 6: Security & Privacy - Privacy - Automation skal inneholde denne tilgangen.

1 Oppgave 1 - Programmering med VS Code

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

1.1 Oppgave 1.1 - Mitt første prosjekt i VS Code

1. Åpne VS Code. En god måte å gjøre dette på er å trykke **Cmd+Space** og skrive **code**. På toppen av listen vil det stå **Visual Studio Code** og når du velger alternativet vil programmet starte.
2. Lag en ny tom mappe et valgfritt sted (f.eks. i **Documents**-mappen din). Gå inn i VS Code og trykk på **File** øverst til venstre, deretter trykk **Open**. Velg mappen du laget. For at programmet skal kjøre senere kan det **ikke være mellomrom** i mappenavnet eller mapper høyere opp i hierarkiet. Det er god praksis å bruke bindestrek istedenfor mellomrom i mappenavn og filnavn. *Ved eventuelle spørsmål du får om at VS Code ønsker tilgang til mapper på din Mac svarer du **Yes**.*
3. Trykk **Cmd+Shift+P** for å få opp *kommandoboksen*. Begynn å skrive **TDT4102: Create Project from TDT4102 Template** og trykk **Enter** når du har markert valget. Velg deretter **Blank project**. Se **Figur 7** for hvordan kommandoboksen ser ut etter å ha skrevet **tdt**.



Figur 7: Kommandoboksen (Command Palette) etter å ha skrevet **tdt** første gang.

4. Til venstre i VS Code vil du nå se en oversikt over filene i prosjektmappen din. Her skal det ligge tre elementer: filene **main.cpp** og **Makefile**, og mappen **.vscode**.
Når tillegget gjenkjenner et korrekt oppsatt prosjekt vil det stå **TDT4102 Project** ✓ nederst i venstre hjørne av VS Code. For å sjekke at du har åpnet VS Code riktig og at alt er som det skal kan du alltid se etter denne teksten og haken når du har åpnet et prosjekt som er korrekt oppsatt for øvingsopplegget. Det vil for eksempel ikke være synlig i en tom mappe.
5. Filen **Makefile** og mappen **.vscode** inneholder innstillinger som bestemmer hvordan VS Code skal kompilere og kjøre koden i prosjektet ditt. Vi anbefaler at du ikke endrer disse filene. Hvis du mot formodning skulle endre noen av disse filene kan du gjøre **Punkt 3**

om igjen og velge **Overwrite** for alle spørsmålene du får, bortsett fra når du får spørsmål om å skrive over `main.cpp`.

6. 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.
7. Kjør programmet ditt ved å trykke **Ctrl+F5** (det kan være du må trykke **fn**-tasten for å få tilgang til F-tastene, det blir i så fall **fn+Ctrl+F5**), eller ved å gå inn i menyen **Run** og velge **Run 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.

(Det opprettes mapper i prosjektet som heter **Debug** og **Release** når programmene bygges. Disse mappene inneholder det byggede programmet for hhv. debug-versjon og release-versjon av programmet. Enn så lenge kan disse mappene ignoreres.)

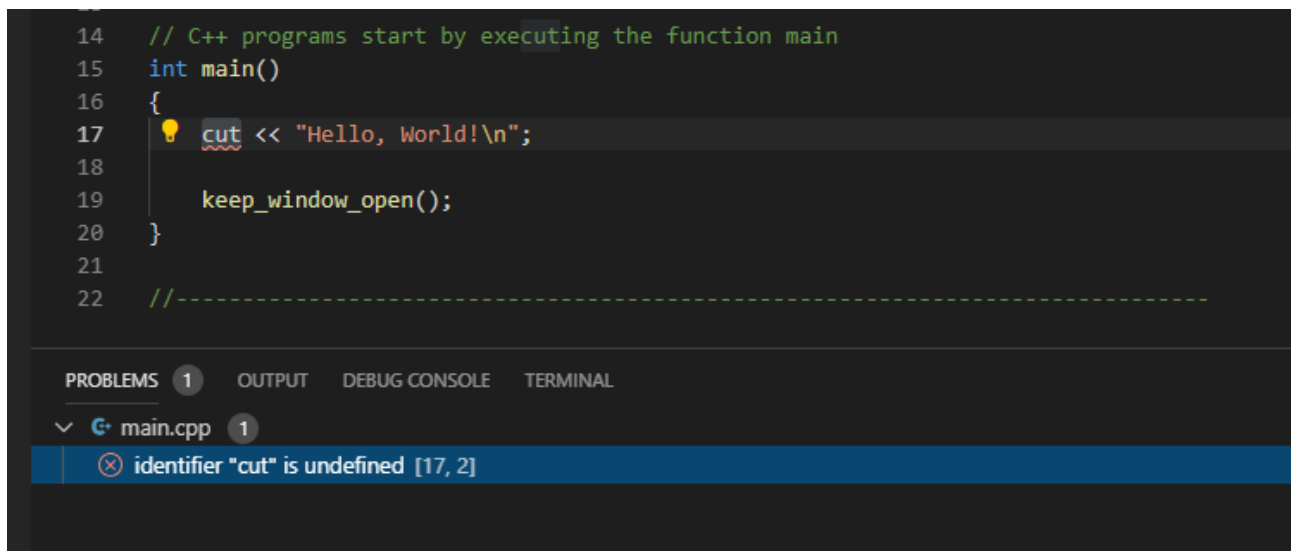
Om du bruker Mac vil ikke dette vinduet automatisk få fokus. Se om det har dukket opp i bakgrunnen om du ikke ser det.

1.2 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 (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. Slett 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 krøllparentesene, skrive `cout` som `cut` osv.
4. Det skal også dukke opp feilmeldinger i Problems-fanen nederst i VS Code, se [Figur 8](#). 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 8: Problems-fanen i VSCode

1.3 Oppgave 1.3 - Eksempelprogrammer fra forelesningene

Ved å åpne kommandovinduet (`Cmd+Shift+P`) og skrive `TDT4102: Create Project from TDT4102 Template` vil du få opp valg om å opprette et nytt prosjekt. Til nå har vi brukt malen `Blank project`. Det er også to valg som heter `Examples` og `Configuration only`. Ved å velge `Examples` vil det dukke opp en liste med eksempelprogrammene fra forelesningene. Disse kan åpnes på samme måte som andre maler. Eksempelene inneholder ikke prosjektinnstillinger som `.vscode`-mappen eller `Makefile`. Disse prosjektinnstillingene finnes i malen `Configuration only` og den bruker vi for å gjøre eksempelprogrammene komplette slik at de kan kjøres fra VS Code.

For å åpne og kjøre et eksempel vil vi:

1. Lage en ny mappe og åpne den i VS code som beskrevet i oppgave 1.1.2
2. Opprette et prosjekt med malen `Configuration only` (`.vscode` og `Makefile`)
 - `Cmd+Shift+P` og velg `TDT4102 Create Project from TDT4102 Template`
 - Velg `Configuration Only`
3. Åpne en av malene i `Eksempler` mappen.
 - `Cmd+Shift+P` og velg `TDT4102 Create Project from TDT4102 Template`
 - Velg `Examples`
 - Skriv inn `hello_graphics` i søkefeltet og velg eksempelet
4. Kjør eksempelprogrammet ved å trykke `Ctrl+F5` (det kan være du må trykke `fn`-tasten for å få tilgang til F-tastene, det blir i så fall `fn+Ctrl+F5`), eller ved å gå inn i menyen **Run** og velge **Run Without Debugging**.

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.

1.4 Oppgave 1.4 - Lage .zip for innlevering

Tillegget TDT4102 Tools har en funksjon for å pakke filene i øvingen til en `.zip`-fil som kan leveres på BlackBoard. Trykk på `Cmd+Shift+P` og skriv `TDT4102: Prepare a zip file for delivery`. Trykk enter. En `.zip`-fil med navn `Handin.zip` vil dukke opp i margin til venstre og ligge i mappen med øvingen. Du må selv laste opp filen på BlackBoard for å få godkjent øvingen.

NB! DU skal ikke levere noe for denne øvingen, men test ut at det fungerer å lage en `.zip`-fil.

1.5 Oppgave 1.5 - Kompilering fra kommandolinje (frivillig oppgave)

Denne oppgaven er ikke nødvendig for å følge øvingsopplegget videre, men gir bedre innsikt i hvordan kompilering og kjøring av C++ kode fungerer, uten at Visual Studio Code gjør det for deg.

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 9**.

```
#include <iostream>
int main() {
    std::cout << "Hello World" << std::endl;
}
```

Figur 9: Eksempelprogram uten avhengigheter på Graph.h eller Simple_window.h

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 stil og format i koden. Skriv inn eksemplet fra Figur 9. 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 directory). 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` (`list`) 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 i så fall 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.

2 Feil og andre meldinger under installering

Denne seksjonen oppdateres ettersom vi blir gjort oppmerksom på problemer under installasjon.