Norges teknisk–naturvitenskapelige universitet
Institutt for datateknologi og informatikk

TDT4102 Prosedyre- og objektorientert programmering Vår 2022

Øving 0 for GNU/Linux

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
- Bli kjent med infobanken

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 2022 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.

Hvis du kjører GNU/Linux eller en annen form for Unix-liknende operativsystem (macOS har egen øving 0), er denne øvingen en generell veiledning til hvordan fagets øvinger *kan* bygges og kjøres. For å gjøre det enkelt er det i denne øvingen også kun installering i Ubuntu som vil vises, det antas at de som velger å benytte andre varianter av Linux kjenner til hvordan et bibliotek lastes ned, kompileres og installeres. **Merk at de to primære platformene som anbefales og støttes i faget TDT4102 er Windows PC og Mac.** De aller aller fleste studentene i faget benytter PC eller Mac, og *fagstaben har ikke kapasitet til å gi veiledning i bruk av Linux.* Hvert år har vi et mindre antall studenter i faget som er godt kjent med Linux, og dette notatet er ekstra hjelp til disse, selv om disse studentene pleier å klare seg selv. Notatet er *ikke* en oppfordring til PC eller Mac brukere til å skifte over til GNU/Linux.

Programvare som installeres i denne øvingen er bl.a. `clang`, `make` og `FLTK`. *Denne øvingen tar utgangspunkt i Ubuntu 18.04 LTS.*

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.

Oppgave 0 - Installasjon av byggeverktøy og FLTK

Linjen under installerer bl.a. make og clang.

```
sudo apt install make cmake clang git libjpeg-dev libx11-dev
```

Installering av FLTK 1.4

Kildekoden til FLTK 1.4 lastes ned fra github vha kommandoen

```
git clone git@github.com:fltk/fltk.git
```

Mappen `fltk` inneholder nå kildekoden til FLTK, og biblioteket skal bygges og installeres. Det gjøres vha. kommandoene under.

```
cd fltk
mkdir build
cd build
cmake .. # Genererer makefiler
make # Bygger FLTK
sudo make install # Installerer FLTK
```

FLTK 1.4 er versjonen faget bruker. Vi kjenner til at FLTK 1.3 kan installeres gjennom ***`apt`****, men vi anbefaler å bruke versjon 1.4 for å unngå uventede forskjeller. Fagets extension til VS Code krever FLTK 1.4, så hvis du vil bruke den må du installere versjon 1.4.* Sannsynligvis vil det gå greit å bruke FLTK 1.3, men det anbefales å installere FLTK 1.4 vha. metoden over for at øvingene skal oppføre seg som forventet. FLTK 1.3 kan installeres i Ubuntu vha.

```
sudo apt install libfltk1.3 libfltk1.3-dev
```

Installering av `graph_lib`

Boken og øvingene bygger på en headerfil og en wrapper for et grafikkbibliotek. Headerfilen `std_lib_facilities.h` vil skjule noe av kompleksiteten og gjøre programmering i C++ tryggere den første halvdelen av kurset. Wrapperen `graph_lib` gjør det mulig å bytte ut det underliggende grafikkbiblioteket (fltk) uten at applikasjoner som benytter `graph_lib` må kompileres på nytt eller endres.

Last ned siste versjon av `linux-install.zip` fra fagets gitlab. Filen heter f.eks. `linux-install-2.0.0.zip` det øyeblikket oppgaveteksten er skrevet. Pakk ut innholdet og gå til roten av den nye stien:

```
unzip linux-install-2.0.0.zip
cd install
```

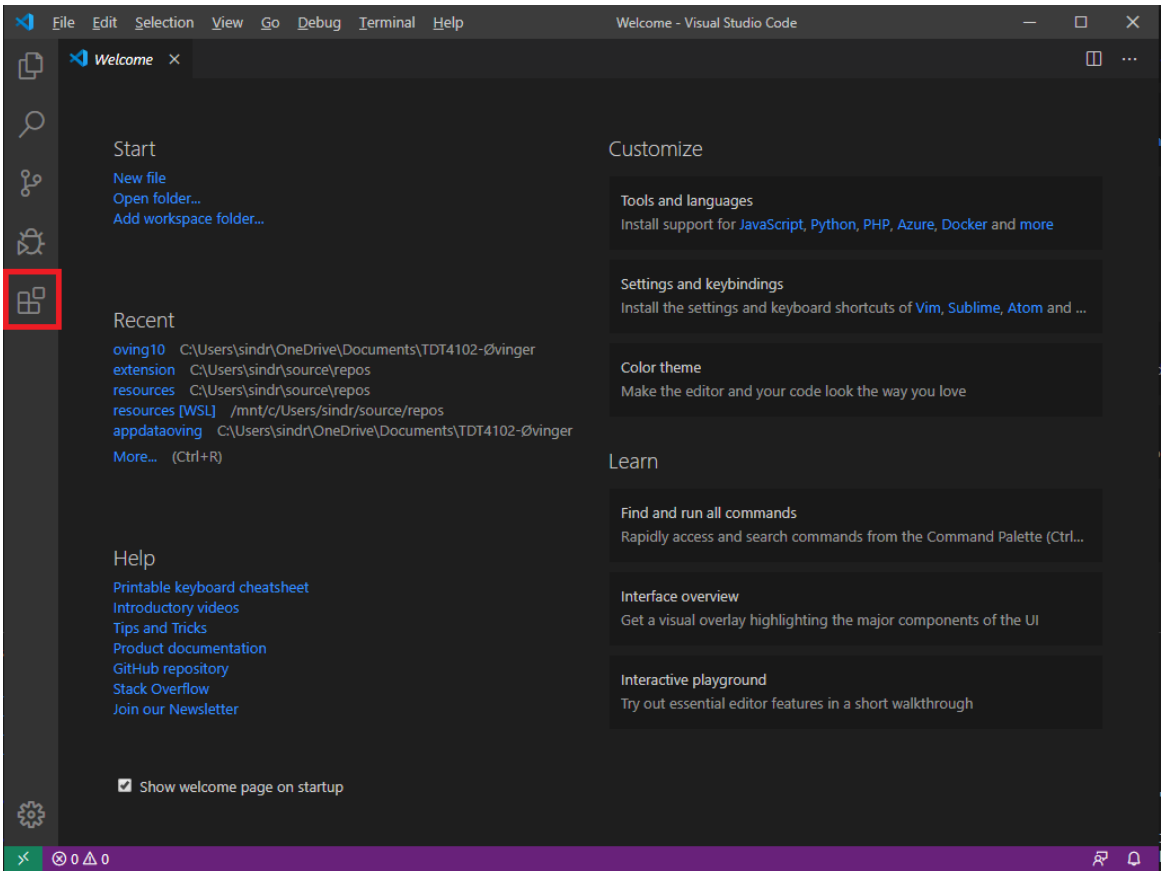
Du bestemmer selv hvor du ønsker å plassere headerfiler og bibliotek. Et godt alternativ for de fleste er å kompilere `Graph_lib` og legge det i et bibliotek, som installeres til `/usr/local`. Det følger med en `Makefile` som gjør dette for deg. *Fagets extension til VS Code krever at .lib-filene ligger i `/usr/local/lib/Graph_lib`.*

```
cd Graph_lib
make all
sudo make install
cd ..
```

Nå skal alle bibliotek være installert og du er klar for å installere VS Code. Hvis du ikke ønsker å bruke VS Code skal det være greit å bruke, eller ta inspirasjon fra, Makefilen vi har skrevet. Den finnes på fagets gitlab, https://gitlab.stud.idi.ntnu.no/tdt4102/vs-code/resources/-/blob/master/linux/common/Makefile. I korte trekk linker den mot Graph_lib og FLTK og definerer diverse mål som gjør VS Code-delen av oppsettet ryddigere.

Oppdatering av Graph_lib

Hvis fagstaben velger å oppdatere Graph_lib underveis i semesteret vil vi legge ut en oppdatering på gitlab. Oppdateringen kan installeres ved å gjøre samme steg som da du installerte Graph_lib i første omgang. Det blir nok ingen store endringer underveis, så her er det forhåpentligvis kun snakk om enkle bugfiks. Vi skal prøve å kommunisere eventuelle oppdateringer av installeringsfilene på f.eks. Blackboard.



Figur 1: VS Code Extension-meny i rødt.

Installering av VS Code

Installeringsveiledning finnes på https://code.visualstudio.com/docs/setup/linux. For Ubuntu går den i korte trekk ut på å laste ned og kjøre `https://go.microsoft.com/fwlink/?LinkID=760868`.

Etter at du har installert VS Code kan du gå til Oppgave 1 for å lære hvordan du kan sette opp dine egne prosjekter.

0.1 Installere TDT4102 Tools

VS Code er en teksteditor med flere nyttige funksjoner. I dette faget har vi laget et tillegg (extension) til VS Code som gjør det lettere å holde maler (templates) oppdatert. Når fagstaben oppdaterer malene vil de automatisk lastes ned av tillegget. Windows- og Mac-brukere har i tillegg installering av kompilator og andre byggeverktøy i extensionen. For dere med GNU/Linux er ikke det like rett fram siden flere har ulik distro, personlige preferanser osv. Vi ønsker ikke å ta for hard kontroll over oppsettene deres, så her er det litt mer manuelt arbeid å holde Graph_lib oppdatert - vi antar at vi ikke må oppdatere Graph_lib underveis i semesteret.

1. Åpne VS Code og velg Extensions fra menyen til venstre, se Figur 1. (`Ctrl+Shift+X`)
2. Søk etter `TDT4102 Tools`, og trykk på `Install`.

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.

Merk: Insider-version
VS Code kommer til å spørre om du vil bli med i insider-programmet. Takk **nei** til det.

1 Oppgave 1 - Programmering med VS Code

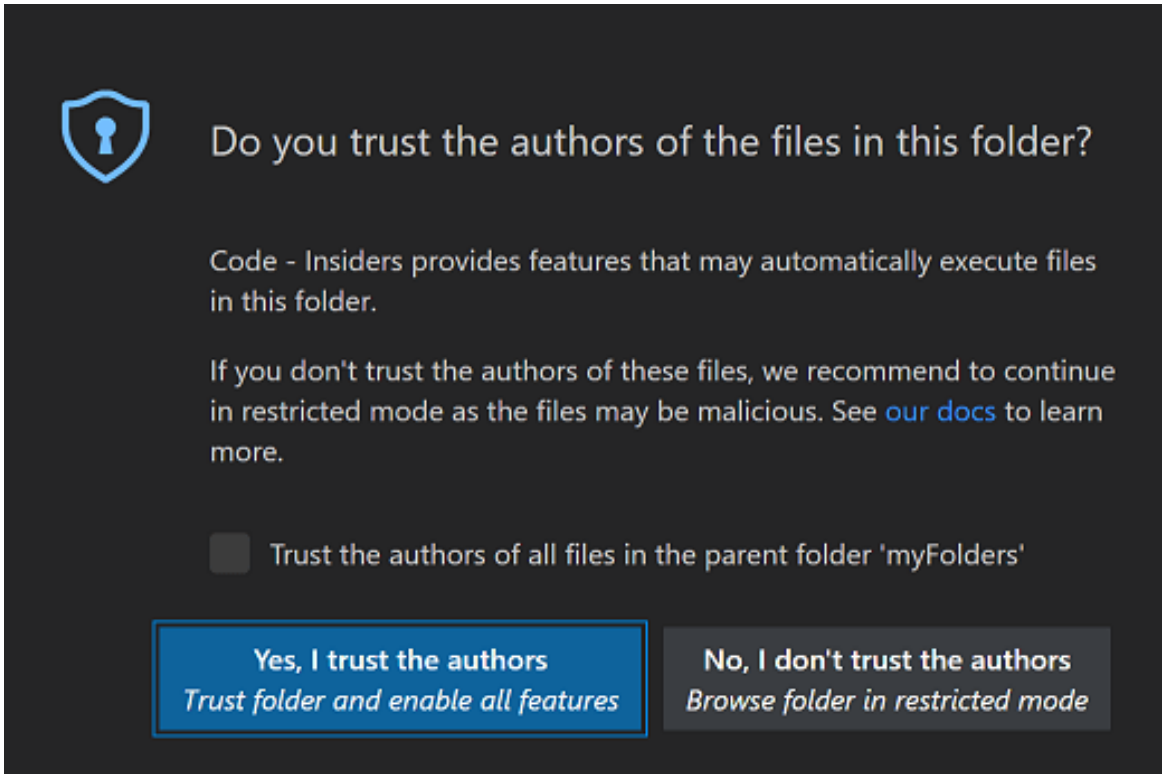
I denne oppgaven antas det at du allerede har installert VS Code, som beskrevet i Oppgave 0.

1.1 Oppgave 1.1 - Mitt første prosjekt i VS Code

1. Åpne VS Code som du normalt åpner programmer. I Ubuntu vil en standard-installasjon la deg trykke Windows-/Super-knappen og skrive `code`, da vil du få mulighet til å starte Visual Studio Code.

2. Lag en ny tom mappe et valgfritt sted. Gå inn i Visual Studio Code og trykk på `File` øverst til venstre, deretter trykk `Open folder`. Velg mappen du laget.

   Hvis du blir spurt om du stoler på mappen 2, trykk på `Yes, I trust the authors`. For mer info klikk her.

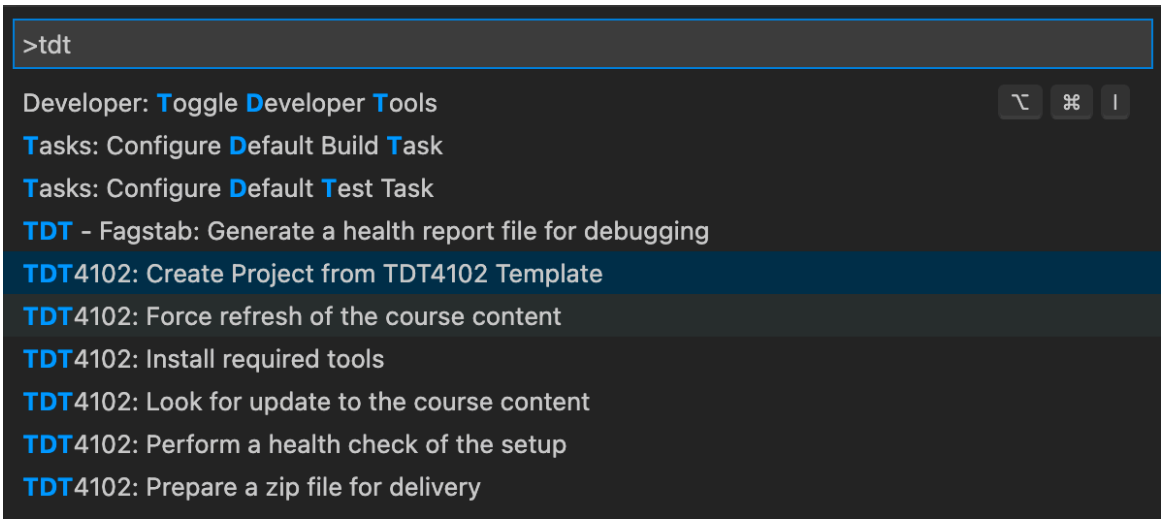


Figur 2: Meny hvor du kan gi `tillit` til arbeidsmappen.

3. Trykk `Ctrl+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 3 for hvordan kommandoboksen ser ut etter å ha skrevet `tdt`.

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



Figur 3: Kommandoboksen (`Command Palette`) etter å ha skrevet `tdt` første gang.

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 skrive `Ctrl+Shift+P`, skrive `TDT4102: Create Project from TDT4102 Template` og velg `Configuration only`. Alternativt 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 som en fane i VS Code så du kan se og redigere den.

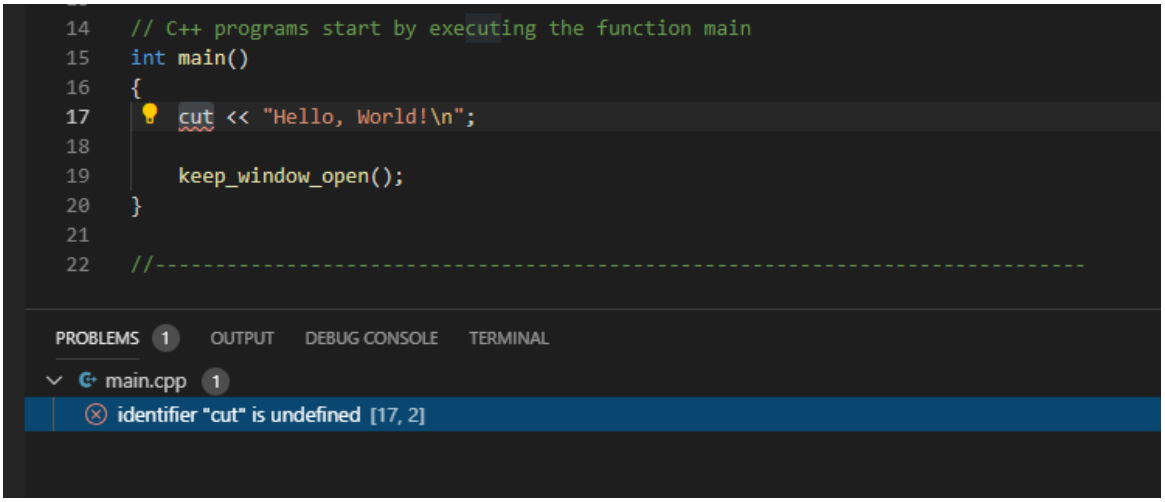
7. Kjør programmet ditt ved å trykke `Ctrl+F5` (det kan være du må trykke `fn`-tasten for å få tilgang til F-tastene, det blir i så fall `fn+Ctrl+F5`), eller ved å gå inn i menyen **Run** og velge **Run Without Debugging**. I terminalvinduet på bunnen (den kan flyttes) av `VS Code` vil teksten "Hello, World!" være skrevet. Hvis du ikke kan se terminalen trykk på Terminal i menyen øverst, trykk så New Terminal og kjør programmet igjen. (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.)

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 4. 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 4: Problems-fanen i VSCode

NB: Det er ikke alltid Problems-fanen er pålitelig

Problems-fanen er fin for å fremheve syntaksfeil i koden din, når den fungerer. Det finnes dessverre tilfeller der fanen enten ikke fanger opp alle feil, eller at den til og med uthever ikke-feil i koden din.

Fasiten får du alltid ved kompilering av koden din, som skjer når du *bygger* prosjektet med `Ctrl+Shift+B`, eller ved kjøring av programmet. Alle feil du eventuelt har i koden din vises da som feilmeldinger i *terminalen* - denne er alltid pålitelig. Ta derfor Problems-fanen alltid med en klype salt.

1.3 Oppgave 1.3 - Eksempelprogrammer fra forelesningene

Ved å åpne kommandovienduet (`Ctrl+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. Eksemplene inneholder ikke prosjektinstillinger 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`)
   - `Ctrl+Shift+P` og velg `TDT4102 Create Project from TDT4102 Template`
   - Velg `Configuration Only`
3. Åpne en av malene i `Eksempler` mappen.
   - `Ctrl+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 blå sirkel. 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 `Ctrl+Shift+P` og skriv `TDT4102: Prepare a zip file for delivery`. Trykk enter. En .zip-fil med navn `Handin.zip` vil dukke opp i margen 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 - Infobanken

En av funksjonene til TDT4102 Tools gir det tilgang til fagets infobank. Her vil du kunne få tilgang på nyttige ressurser som artikler, videoer og øvinger. Formålet er at du enkelt skal kunne finne det du trenger inne i `VS Code`.

For å åpne infobanken trykk på `Ctrl+Shift+P` og skriv `TDT4102: Open infobank`. En liste over ulike ressurser vil dukke opp hvor du kan søke eller bla/scrolle til den ressursen du ønsker å bruke. Du kan søke både etter tittel eller nøkkelord.

Prøv å åpne følgende ressurser:

1. `Piazza` - fagets forum
2. `Graph_lib dokumentasjon`
3. `Feilmeldinger` - en artikkel som hjelper med debugging
4. `Hello graphics` - her finner du lenke til en video som forklarer hvordan man lager grafikk i c++
5. `Øving 0` - Du kan til og med finne denne øvingen i infobanken! Øvingene i faget vil bli lagt ut i infobanken i løpet av året

1.6 Oppgave 1.6 - 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 funker, 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 velger å ikke benytte *Graph.h* eller *Simple_windows.h* i dette kommandolinje-eksempelet, ettersom det er mer avansert å kompilere "for hånd". Eksempelprogrammet vi skal bruke i denne oppgaven vises i Figur 5.

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

Figur 5: Eksempelprogram uten avhengigheter på Graph.h eller Simple_window.h

For å bygge et program fra terminalen kan du bruke `clang++`. Det kan gjøres på følgende måte:

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

Da vil du få en fil i samme mappe som heter `a.out`. For å kjøre programmet fra terminalen skriver du:

```
./a.out
```

Står det "Hello World" i terminalen er du i mål.

2 Feil og andre meldinger under installering

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

2.1 Bruk av skylagring og antivirus

Hvis du dukker opp i problemer, kan dette være på grunn av bruk av skylagringstjenester (iCloud, OneDrive, Dropbox, etc.) Sørg for at skylagringstjenestene ikke virker på arbeidsfilene.

3 Nyttig å vite

3.1 Auto save

Vi anbefaler sterkt å bruke auto save i dette faget. For å skru det av og på gå til menyen øverst og trykk på File så på `Auto save`. Hvis Auto Save er av vil TDT Tools minne deg om å skru det på.