

## i Informasjon

Institutt for datateknologi og informatikk

### Inspiraøving 1 TDT4102 - Prosedyre- og objektorientert programmering

Hjelpemiddelkode/Tillatte hjelpemidler: A / Alle hjelpemidler tillatt

#### ANNEN INFORMASJON:

Dette er en øving som skal forberede dere og gi dere et inntrykk av hvordan eksamen kan være. Det er ingen minuspoeng, og vi anbefaler å svare på alle oppgavene etter beste evne. Selv oppgaver som ikke er fullstendig besvart, kan gi uttelling, så skriv det du klarer.

Les gjennom hele oppgavesettet før du begynner å løse oppgaven. Disponer tiden godt! I noen oppgaver vil det være naturlig å bruke funksjoner fra tidligere oppgaver, disse kan brukes som om de fungerer selv om du ikke fikk til den tidligere oppgaven.

Etter at øvingene er ferdig rettet, vil dere se resultatet på BlackBoard. Øvingen er som andre øvinger Bestått/Ikke bestått.

Gjør dine egne antagelser og presiser i besvarelsen hvilke forutsetninger du har lagt til grunn i tolkning/avgrensing av oppgaven.

**Lagring:** Besvarelsen din i Inspira Assessment lagres automatisk. Jobber du i andre programmer – husk å lagre underveis.

**Juks/plagiat:** Øvingen skal være et individuelt, selvstendig arbeid. Videre er det tillatt å bruke hjelpemidler, men merk at alle besvarelser blir kontrollert for plagiat.

**Tekniske problemer:** Ved tekniske problemer som ikke er relatert til oppgavene. Vennligst ta kontakt med undervisningsassistenter (undass) på Teams, eller ved å sende en mail til: [tdt4102-undass@idi.ntnu.no](mailto:tdt4102-undass@idi.ntnu.no)

#### Filopplasting:

**Alle filer må være lastet opp i besvarelsen før tida går ut.** Det er lagt til 15 minutter for å laste opp prosjektmappen igjen. Tilleggstida inngår i gjenstående tid som vises øverst til venstre på skjermen.

I andre del av øvingen skal du laste ned en prosjektmappe og åpne denne i VS-code. Du må vite hvordan du laster ned, pakker ut, og setter opp mapper fra en zip-fil som VS-code (eller tilsvarende) C++ prosjekter på maskinen din. Til slutt må du være i stand til pakke alle slike mapper sammen i en zip-fil igjen for å levere det du har kodet, innen tidsfristen, for å kunne bestå denne øvingen.



## Consent

### REGLER OG SAMTYKKER

Dette er en **individuell** øving. Du har ikke lov til å kommunisere (gjennom web-forum, chat, telefon, hverken i skriftlig, muntlig eller annen form), ei heller samarbeide med noen andre under eksamen.

**Før du kan fortsette til selve øvingen må du forstå og SAMTYKKE i følgende:**

**Under øvingen:**

**Jeg skal IKKE motta hjelp fra andre.**

☐ Aksepter

**Jeg skal IKKE hjelpe andre eller dele løsningen min med noen.**

☐ Aksepter

**Jeg skal IKKE copy-paste noe kode fra noen eksisterende online/offline kilder. (Du kan se, og deretter skrive din EGEN versjon av koden).**

☐ Aksepter

**Jeg er klar over at øvingen kan bli underkjent uavhengig av hvor korrekt svarene mine er, hvis jeg ikke følger reglene og/eller IKKE aksepterer disse utsagnene.**

☐ Aksepter

# 1 Teori 1

Funksjonen `readFromFile()` skal lese heltall fra en strukturert fil og lagre de i en vektor ved hjelp av `getline()`. Filen inneholder linjer med komma-separerte heltall. Hva er feil i koden?

```
1  vector<int> readFromFile(string filename) {  
2      ifstream ist{filename};  
3      if (!ist){  
4          throw "Could not open file "+filename;  
5      }  
6      vector<int> numbers;  
7      int number;  
8      while (getline(ist, number, ',')){  
9          numbers.push_back(number);  
10     }  
11     return numbers;  
12 }
```

Velg ett alternativ:

- ☐ Funksjonen kan ikke kaste en string som unntak.
- ☐ Funksjonen kan ikke returnere en vector som er opprettet inne i funksjonen.
- ☐ Funksjonen kan ikke dytte mer enn en linje med input til numbers.
- ☐ Funksjonen kan ikke bruke `getline` for å lese et tall til en `int`.

---

Maks poeng: 1

## 2 Teori 2

Gitt koden under:

```
1 class Example {
2     public:
3         Example(int a) {
4             // Code for constructor one
5         }
6         Example(int a, int b) : Example{a} {
7             // Code for constructor two
8         }
9     }
```

Hva kalles denne måten å implementere konstruktører på?

- ☐ Implisitt konstruktør
- ☐ Default konstruktør
- ☐ Dobbel konstruktør
- ☐ Delegerende konstruktør

Hva er fordelen med denne teknikken?

Velg ett alternativ

- ☐ Mindre binærkode kompleksitet
- ☐ Automatisk syntaks sjekking
- ☐ Kjøretids advarsler
- ☐ Mer gjenbruk av kode

---

Maks poeng: 2

### 3 Teori 3

Se koden under:

```
1  int i = 1;
2  while (true) {
3      if (i++ > 10) {
4          break;
5      }
6
7      if (i % 2) {
8          cout << "Fizz" << " ";
9      } else {
10         continue;
11         cout << "Buzz" << " ";
12     }
13     cout << endl;
14 }
```

Når programmet kjøres skriver det ut: Fizz Fizz Fizz Fizz Fizz

Men, intensjonen til programmereren var å skrive Fizz og Buzz annenhver gang, slik: Buzz Fizz Buzz Fizz Buzz ..

Hva må du endre i koden for å fikse den?

**Velg ett alternativ:**

- ☐ Fjern “break” (linje 4).
- ☐ Fjern “continue” (linje 10).
- ☐ Flytt plasseringen av linje 11: cout << "Buzz" << " ";
- ☐ Endre betingelsen i while-løkken.

---

Maks poeng: 1

## 4 Teori 4

Se på følgende kode:

```
1  enum class Animal {
2      COW = 0, GOAT, HORSE, DOG
3  };
4  int main() {
5
6      Animal a = 2;
7
8      switch(a){
9          case Animal::COW:
10             cout << "cow";
11             break;
12          case Animal::GOAT:
13             cout << "goat";
14             break;
15          case Animal::HORSE:
16             cout << "horse";
17             break;
18          case Animal::DOG:
19             cout << "dog";
20             break;
21     }
22     cout << endl;
23
24     return 0;
25 }
```

Du vil at koden skal skrive "horse" og tilordner derfor 2 til variabelen siden det h rer til "HORSE" i "enum class Animal". Under kompilering innser du at det ikke g r, siden det ikke er mulig   tilordne et heltall til en "enum" variabel.

En opplagt l sning ville v re   bytte ut linje 6 med "Animal a = Animal::HORSE", men som du vet er det alltid flere m ter   gj re ting p  i programmering.

Hvilken annen liten endring kan du gj re i linje 6 slik at koden skriver ut "horse"?

Skriv svaret kort og konsist her:

Maks poeng: 1

## i Last ned prosjektmappe

I resten av  vingen trenger du   laste ned en prosjektmappe som du  pner i VS-code:

Les **Bruk av utdelt prosjektmappe ved inspira ving 1** i BB under mappen inspira ving for videre bruk av prosjektmappen.

I slutten av  vingen laster du opp igjen prosjektmappen som en zip-fil med besvarelsen din.

[Last ned prosjektmappen her](#)

Resten av oppgavesettet er ogs  tilgjengelig p  [PDF](#)

5    **Spørreundersøkelse - a**

De følgende oppgavene implementerer du i *survey2021.cpp/h*

Fagstaben vil kartlegge hvilket forhold studenter har C++. Igjennom en spørreundersøkelse svarte studenter på spørsmålet: "Hvilket adjektiv forbinder du med faget TDT4102: ". Tekstfilen, "survey\_2021.txt", er formatert slik at alle svarene står oppført separert med **ett** mellomrom.

Lag en funksjon *mapSurvey()* som åpner filen og returnerer et *map<string,int>* hvor strengen er svar-adjektivet og heltallet er antall ganger adjektivet er nevnt i undersøkelsen.

Husk å sjekke om du klarer å åpne filen i funksjonen.

Maks poeng: 10

6    **Spørreundersøkelse - b**

Vi ønsker å finne adjektivene som ble svart flest ganger i undersøkelsen.. Funksjonen *sortVector()* tar inn et map som argument og returnerer en *vector<string>*. Vektoren skal inneholde nøklene i mapet (adjektivene) sortert basert på antall forekomster av adjektivet, fra flest til færrest.

Implementer funksjonen *sortVector()*.

Maks poeng: 10

7    **Spørreundersøkelse - c**

For å systematisere resultatene ønsker fagstaben en oversiktlig tekstfil. Implementer funksjonen *surveyResults()* som genererer en fil på følgende format:

Survey results 2021:  
-----  
<adjektiv>, <antall forekomster>  
<adjektiv>, <antall forekomster>  
...

Elementene skal stå i sortert rekkefølge - adjektivet med flest forekomster skal stå først i filen.

Maks poeng: 10

8    **Kjøretøy - a)**

De følgende oppgavene implementerer du i *vehicle.cpp/h*

I denne oppgaven skal du lage et *enum class* Color som skal inneholde fargene *blue, yellow, red, green*. Dersom du ønsker kan du også legge inn flere farger.

Maks poeng: 10

9    **Kjøretøy - b)**

I denne oppgaven skal du deklare og implementere klassen *Vehicle*.

Klassen skal inneholde

- En streng som inneholder navnet på kjøretøyet
- En Color variabel som inneholder fargen på kjøretøyet. Denne skal være tilgjengelig ved arv.
- Et flyttall speed som inneholder nåværende hastighet på kjøretøyet. Denne skal være tilgjengelig ved arv.
- En konstruktør som tar inn navnet, fargen og hastigheten på kjøretøyet og initialiserer dem til disse verdiene.
- En funksjon allowedSpeed. Denne skal sjekke om et kjøretøy kjører i lovlig hastighet. I dette tilfellet ønsker vi å sjekke om kjøretøyet kjører under 1000 km/h. Sørg derfor for at funksjonen returnerer true dersom kjøretøyet kjører under 1000 km/h, og false dersom det er over 1000 km/h.

---

Maks poeng: 10

10    **Kjøretøy - c)**

I denne oppgaven skal vi lage klassen *Car*. Denne skal arve fra *Vehicle*, samt ha litt ekstra funksjonalitet. Klassen skal inneholde det følgende:

- En variabel numberOfWheels. Denne skal inneholde antall hjul bilen har.
- En konstruktør som initialiserer superklassen Vehicle, samt initialiserer numberOfWheels.
- En funksjon changeColor(Color newColor) som endrer fargen definert i klassen Vehicle til fargen newColor.
- En ny versjon av funksjon allowedSpeed(). Årsaken er at det på de fleste veier i Norge ikke er lov å kjøre over 100 km/t. Sørg derfor for at funksjonen returnerer true om bilen kjører under 100 km/t, og false om den kjører fortere.

---

Maks poeng: 10

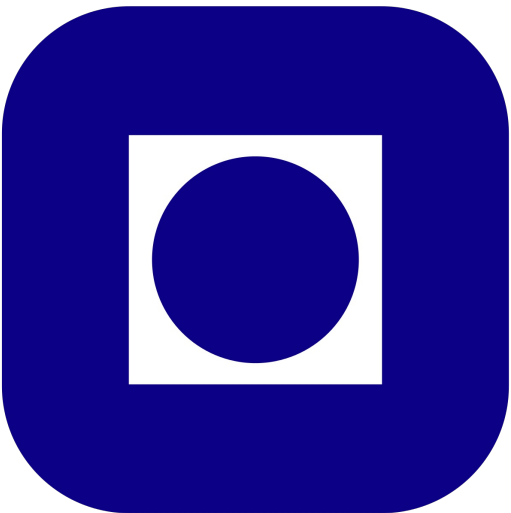


11 NTNU Logo

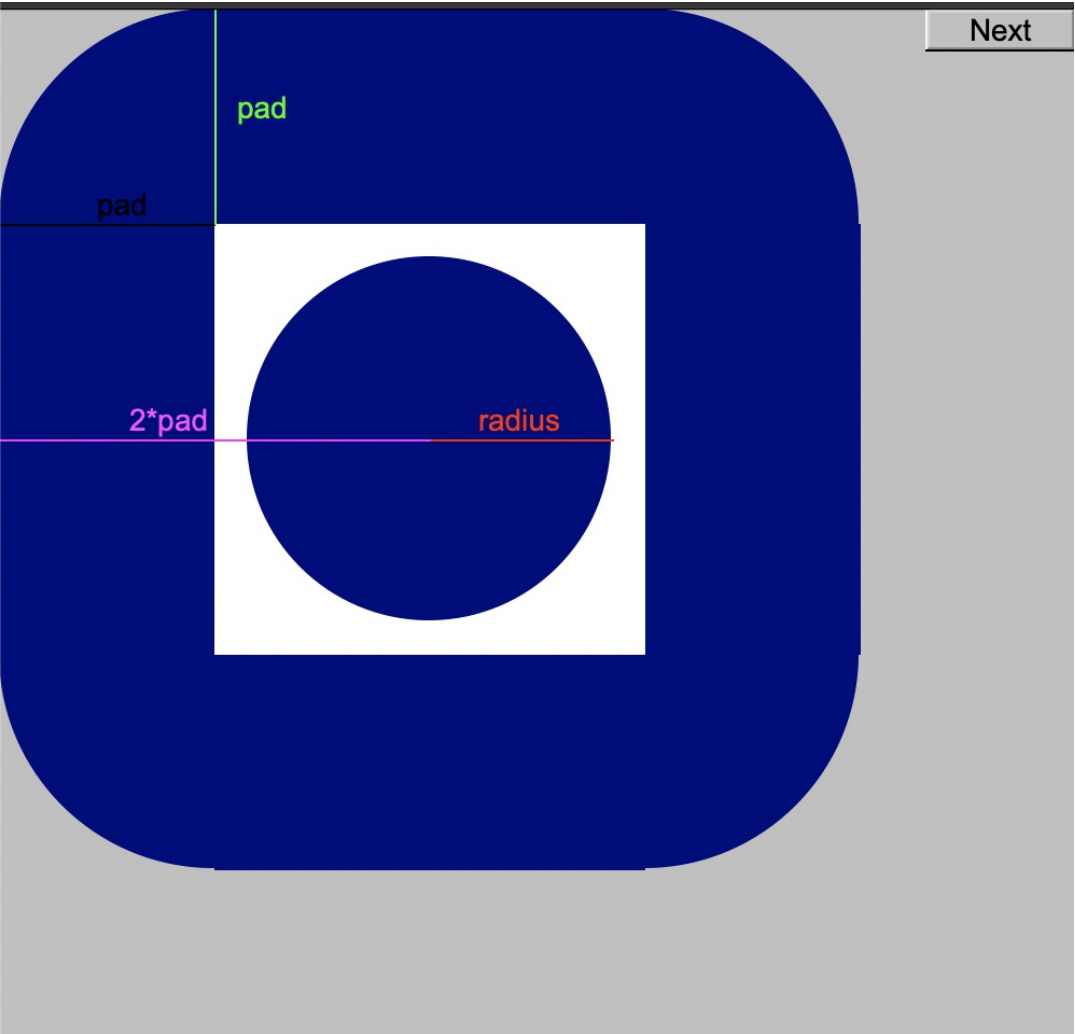
Denne oppgaven implementerer du i *NTNULogo.cpp/h*

Etter et hardt år med korona, så har NTNU bestemt seg for å begynne å spare penger. De bestemmer seg derfor for å bytte ut fancy tegneprogram og dyre fagfolk med FLTK og studenter. **Fullfør implementasjonen av funksjonen *drawNTNULogo()*. Funksjonen skal tegne alle *Shapes* som trengs og plassere dem på korrekt sted.**

Logoen består av to rektangler og fire sirkler i bakgrunnen i tillegg til en rektangel og en sirkel lagt oppå:



Inspirasjon:



Maks poeng: 10

12

Innlevering

Lever prosjektmappen din som en zip-fil.



Last opp filen her. Maks én fil.

Alle filtyper er tillatt. Maksimal filstørrelse er **50 GB**.



Velg fil for opplasting

Maks poeng: 0