



NTNU

# **Hvordan fremme aktiv deltakelse i programmeringsforelesninger (digitale og fysiske) med opptil 500 studenter?**

Fanny Skirbekk, stipendiat ved Institutt for elkraftteknikk (NTNU)

# Aktiv læring og programmering

- Programmering er en utrolig viktig ferdighet.
  - Viktig å sikre godt læringsutbytte fra de grunnleggende progge-fagene.
- **Aktiv læring** => studentene er **aktive deltakere** i læringsprosessen, framfor passive tilskuere
- Aktiv læring gir økt læringsutbytte
- Men aktiv læring er (tilsynelatende) vanskelig å implementere i store forelesninger..

# Bakgrunn

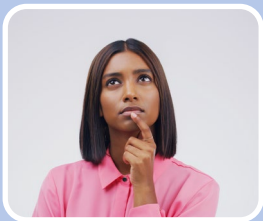
- ***TDT4102 Prosedyre- og objektorientert programmering***
  - Nest største IT-faget på NTNU
  - Studenter fra mange ulike studieprogram
  - Over 1000 oppmeldte studenter
  - Forelesningsparalleller på 500 studenter
- **Hjelpeforelesninger/støtteforelesninger**
- **Undass/vitass i TDT4102 de siste 5 årene**
  - 55 hjelpeforelesninger totalt, hvorav 19 digitale over Zoom

1.



Presenter en oppgave

2.



Gi betenkningstid/diskusjonstid

3.



Gå gjennom løsningen i plenum

# Valg av oppgave

- **Ikke for stor!**
- **Tema som passer forelesningen**
- **Kodeforståelse**
  - Gjerne en gammel eksamensoppgave!
- **Drøftingsspørsmål**
  - Hvorfor bør teknikk X benyttes i tilfelle Y?
  - Hva er fordelene/ulempene med A framfor B?
  - Hva er begrensningene hvis man velger løsning/metode Z?
- **Live-koding basert på student-innspill**
  - Har kun testet for <100 studenter.

# Hvordan presentere oppgaven

- **Skriv oppgaven på lysark**
  - Gjør det enkelt å få den med seg.
  - Kan se på oppgaven flere ganger under «tenketiden».
- **Digitalt: bruk polls (avkrysning)**
  - Hindrer kaos i chatten.
  - Gjør at «alle» enkelt kan delta, selv om det er digitalt
  - Oppdage kunnskapshull/vanlige misforståelser.



# **Eksempler på oppgaver som har vært gitt i TDT4102 våren 2022**

## Oppgave 1g, vår 2017: Hva skrives ut?

```
1  stack<int> intStack;
2  for (unsigned int i = 0; i < 10; i++){
3      intStack.push(i);
4  }
5  intStack.pop();
6  intStack.pop();
7  cout << intStack.top() << ", ";
8  intStack.push(100);
9  cout << intStack.size() << ", ";
10 intStack.pop();
11 intStack.pop();
12 cout << intStack.top();
```



Hvilke krav må x og y oppfylle?

```
1  template <typename T>
2  //T er plugg-inn-typen
3  void mySwap(T& a, T& b){
4      T temp = a;
5      a = b;
6      b = temp;
7  }
8
9  void myFunc(){
10     // x og y defineres
11     mySwap(x, y);
12 }
```

## Hvilken datatype skal vi bruke her?

```
1  ??? theAnswer = 42;
2  ??? oppfantBreakpoint = "Betty Holberton";
3  ??? programmeringErMorsomt = true;
4  ??? ukjent = 'x';
5  ??? boktittel = "The Travelling Cat Chronicles";
6  ??? pi = 3.14;
7  ??? krokodillerErDinosaurer = false;
8  ??? kulesteStudiet = "elkraft";
9  ??? utropstegn = '!';
10 ??? tegnMedDobbeltFnutt = "x";
```

## Hvilken metode bør brukes?

1. En funksjon som tar inn en vektor med heltall, og skriver ut alle elementene til skjerm.
2. En funksjon som tar inn en vektor, og endrer alle negative tall til verdien 0.
3. En funksjon som tar inn to heltall, og returnerer summen av tallene kvadrert.
4. En funksjon som bytter om på to heltallsvariablers verdi.

## Oppgave: opprette tabell

Jeg ønsker å opprette en tabell for å ha oversikt over de 5 fineste ordene jeg vet om. Hvordan kan jeg gå fram for å gjøre dette?

Vi livekoder løsningen sammen.

# Betenkningstid – hvorfor?

- **Sikrer at «alle» deltar aktivt**
  - Ingen garanti, men erfaring tilsier så
- **Flere som svarer**
  - Mindre skummelt når man har diskutert
  - Lettere når man har begynt å tenke på det
- **Flere spørsmål og oppfølgingsspørsmål**
  - «Hva hadde skjedd hvis..»

# Betenkningstid – hvor lang?

- **1-3 min tenketid/diskusjonstid**
  - Rekker å starte tenkeprosessen
  - For kort til pause
  - Informer om tidsbegrensningen!
- **Digitalt: åpne poll når spørsmålet stilles**
  - Studentene svarer når de er klare
  - Men *ikke* gi dem mer enn 1-3 minutter!

# Gjennomgang av oppgaven

- **Bryt ned oppgaven**
  - Kodeforståelse er fint å ta linje for linje
  - Gjør det lettere å forklare og mindre skummelt for studentene å svare
- **Fysiske forelesninger: få studentene til å svare**
  - Vent 10-20 trykkende sekunder ved behov
  - De blir flinkere til å svare når de skjønner at de må
  - Pass på å håndtere feilsvar på en «hyggelig» måte. Det skal være rom for å ta feil!
- **Digitalt: gå gjennom «alene»**
  - Men bruk pollsvar og chat-spørsmål for det det er verdt!
- **Åpne opp for spørsmål**
  - Innblikk i hva som er vanskelig
  - Avdekke kunnskapshull

# «Oppsummering»

- Oppfordrer til å teste oppgave-tenketid-gjennomgang!
- Trolig overførbart til andre fagfelt også

**Innspill? Erfaringer? Spørsmål? Tips?**