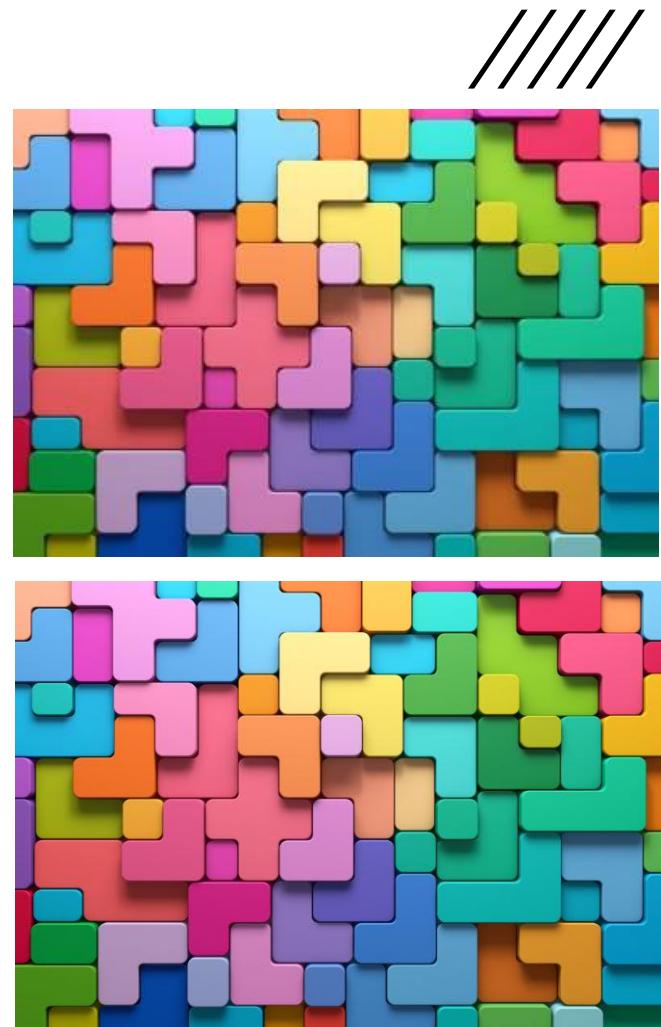


# **OBJEKTER OG KASSER I C++**

FORELESNING 14

FREDAG 3/10

(bilder generert av bing image creator)



## Ekstramateriale om C++

- [Gratis lærebok](#) (gjennom UiO)
- Ikke pensum, men nyttig om man ønsker en ekstra kilde



# Ubesvarte spørsmål fra mandag

- Hvordan kompilere uten makefile
- Fungerer med **#include "array\_list.cpp"** som ikke er best practice (men godkjennes for prosjektet):
  - **g++ test\_array\_list.cpp -o array\_list.exe --std=c++14**
- Fungerer med **#include "array\_list.h"** som er bedre:
  - **g++ test\_array\_list.cpp array\_list.cpp -o array\_list.exe --std=c++14**
- Generelt er det grensesnittet som bør importeres, ikke en bestemt implementasjon av det



# Ubesvarte spørsmål fra mandag

- Hvordan koble til IFI-maskin med **ssh** og **scp**
  - **ssh -J dittbrukernavn@login.uio.no dittbrukernavn@login.ifi.uio.no**
  - **mkdir IN1910-prosjekt2** (i hjemmeområdet: **cd ~**)
  - Sett opp SSH-nøkkel for å bruke **scp**
  - **scp \*.\* dittbrukernavn@scp.uio.no:~/IN1910-prosjekt2**  
(i repo-mappe)
- Mer info her og her



# International English Spelling Chart



United States	Canada	United Kingdom	Australia
color	colour	colour	colour
center	centre	centre	centre
realize	realize	realise/ize	realise
analyze	analyze	analyse	analyse
traveling	travelling	travelling	travelling
defense	defence	defence	defence
computer program, concert program	computer program, concert program	computer program, concert programme	computer program, concert program
gray	grey	grey	grey
fulfill	fulfil(l)	fulfil	fulfil
aging	ag(e)ing	ageing	ag(e)ing





## Til prosjekt 2

- Bruk **-std=c++14** for å bruke C++14-standarden (fra 2014) når man kompilerer
- Husk å teste at koden kompilerer og kjører på en IFI-maskin – det holder ikke at det funker på *din* maskin!
- Enkleste måte er å bruke **scp** til å kopiere kildekoden og **ssh** til å logge inn og kompile / kjøre etterpå

## ● Viktig forskjell på variabel og peker

- En variabel (**Node ny\_node**) ryddes opp når den ikke lenger er definert (f.eks. funksjonen der den defineres er ferdig)
- En peker (**Node\* ny\_node**) ryddes ikke opp av seg selv
- Hvis vi fortsetter å bruke en peker med adressen til en variabel som ikke lenger er definert, kan det være et helt annet objekt der enn vi tror (minnet gjenbrukes når det er ledig!)
- Kompilatoren vil ofte advare oss hvis vi prøver på dette



## ○ Minnelekkasje

- Vi kan lage et objekt direkte med en peker isteden:  
**Node\* ny\_node = new Node();**
- Men da må vi rydde opp selv: **delete ny\_node**
- Hvis vi ikke rydder opp, kan vi få en *memory leak* - vi reserverer mer og mer minne (med **new**) uten å gi beskjed om at disse minneadressene ikke trengs lenger
- (Livekodet eksempel til skrekk og advarsel)



- Lenket liste

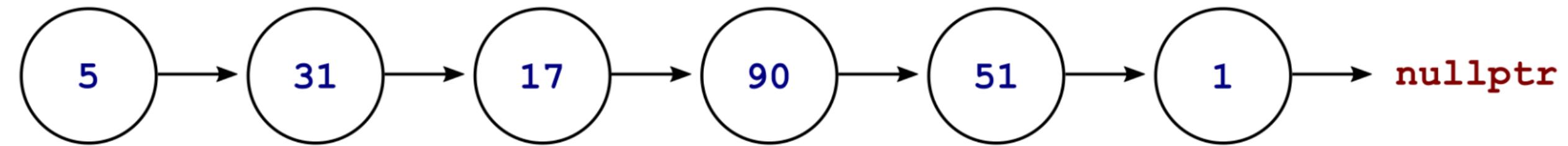


Fig. 33 A simplified drawing of a linked list.



- En lenket liste består av Node-objekter med referanser til neste Node

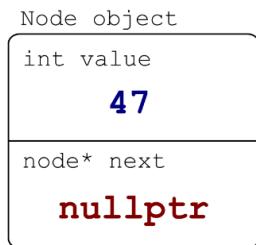


Fig. 31 A single node object. #

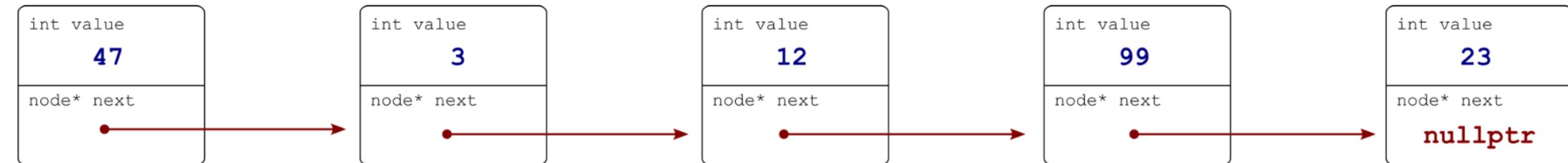
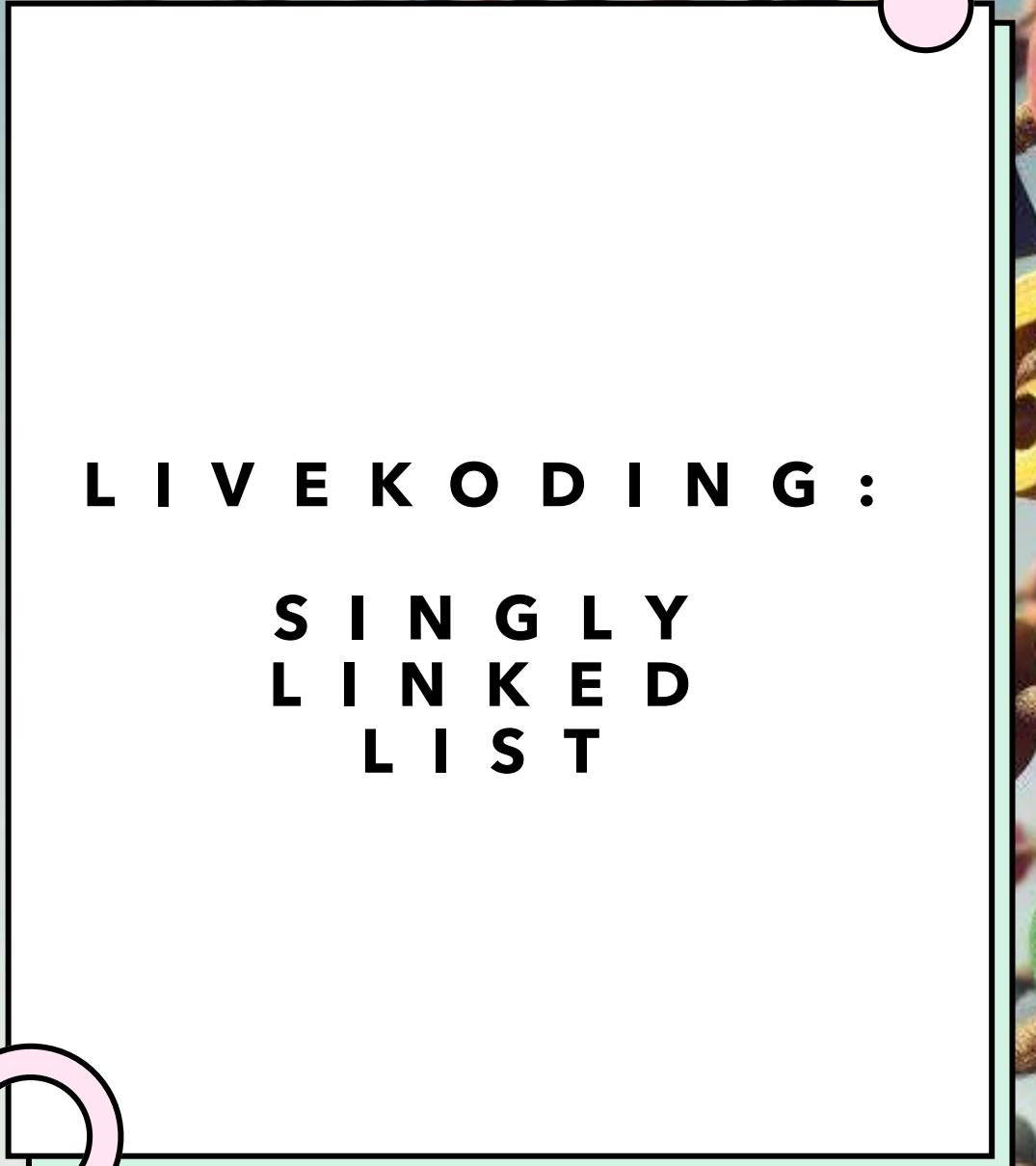


Fig. 32 A chain of node objects. #





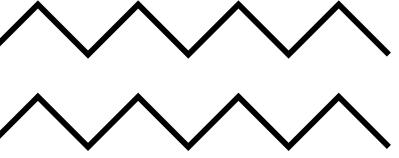
LIVE CODING :  
S I N G L Y  
L I N K E D  
L I S T



## • Forskjell på **struct** og **class**

- Det kan være smart å bruke **struct** når vi bare trenger dataobjekter uten metoder (men de kan ha metoder også), kan sammenlignes med en dataclass i python
- Alt i en **struct** er **public** som standard
- Alt i en **class** er **private** som standard (hvis ikke vi spesifiserer at det skal være **public**)





# Viktig info!

- Ingen forelesninger/gruppertimer neste uke (midtveisuke)
- Vi fortsetter som normalt uken etter (forelesning mandag 13.)
- Midtveisevaluering og eksamensdatoer:  
<https://nettskjema.no/a/549414>

