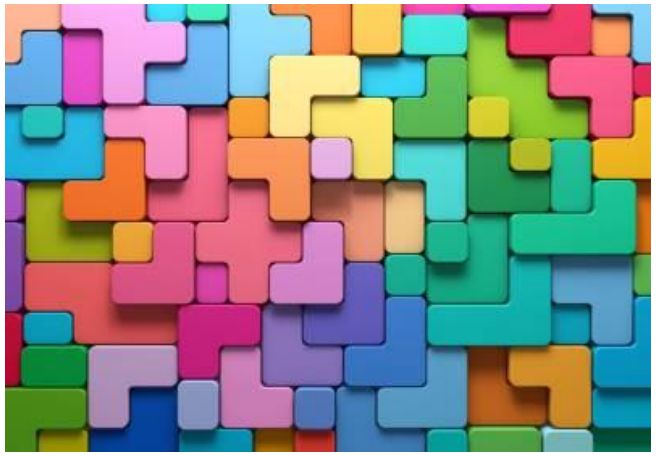
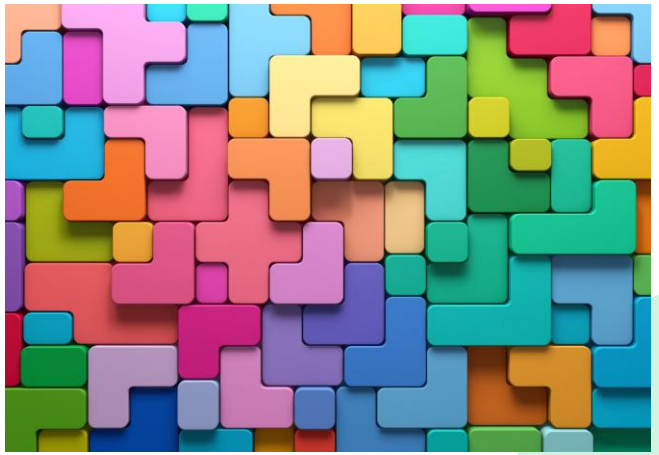
ARRAYS, MINNE OG PEKERE ☞

FORELESNING 12

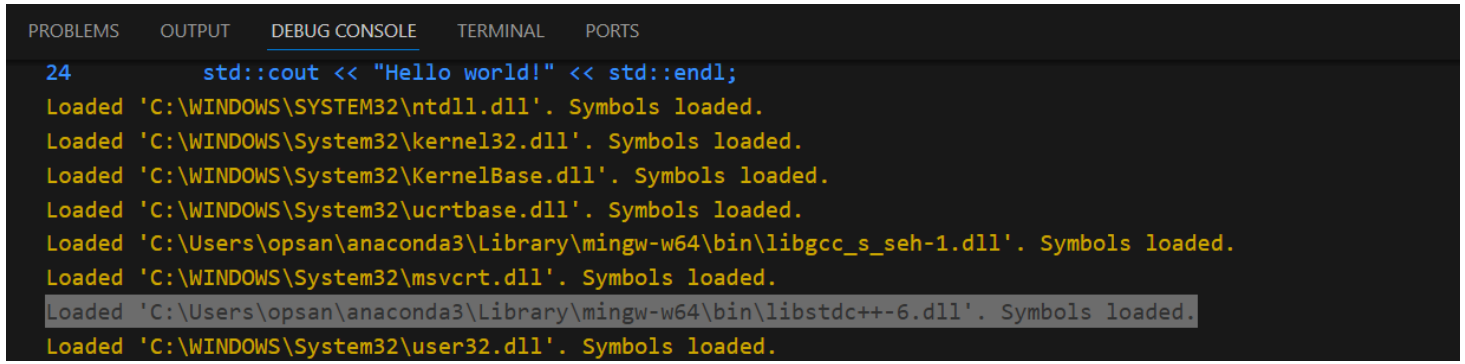
FREDAG 26/9





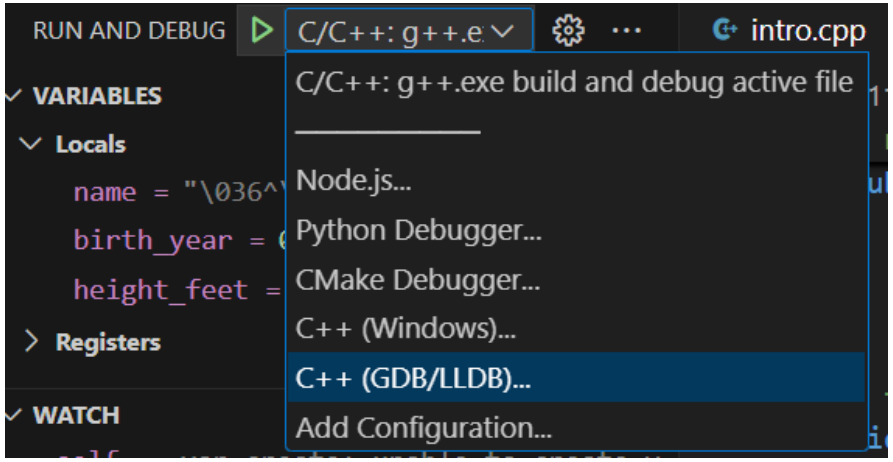
(bilder generert av bing image creator)

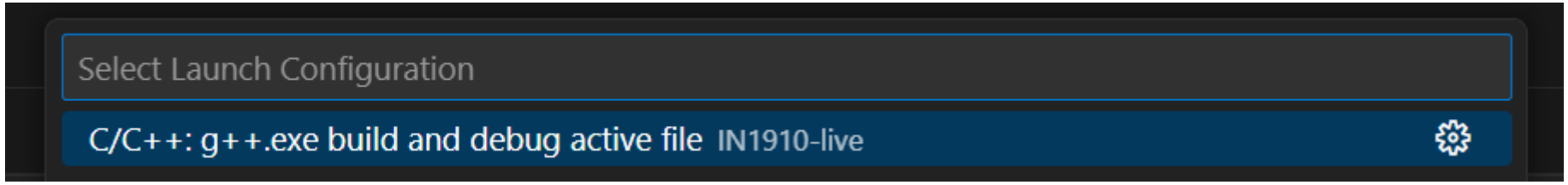
Sjekkliste for diverse trøbbel i Windows



- Legg til **C:\msys64\ucrt64\bin** i PATH under environment variables (system) og start VS Code på nytt
- Kompiler med **–static-libstdc++** (i terminalen)
- Debugger:.vscode → tasks.json legg til **"–static-libstdc++",** under "args"
  (må gjøres for hvert repo, men ikke på hver maskin/bruker)

Debugging i C++ (Windows)





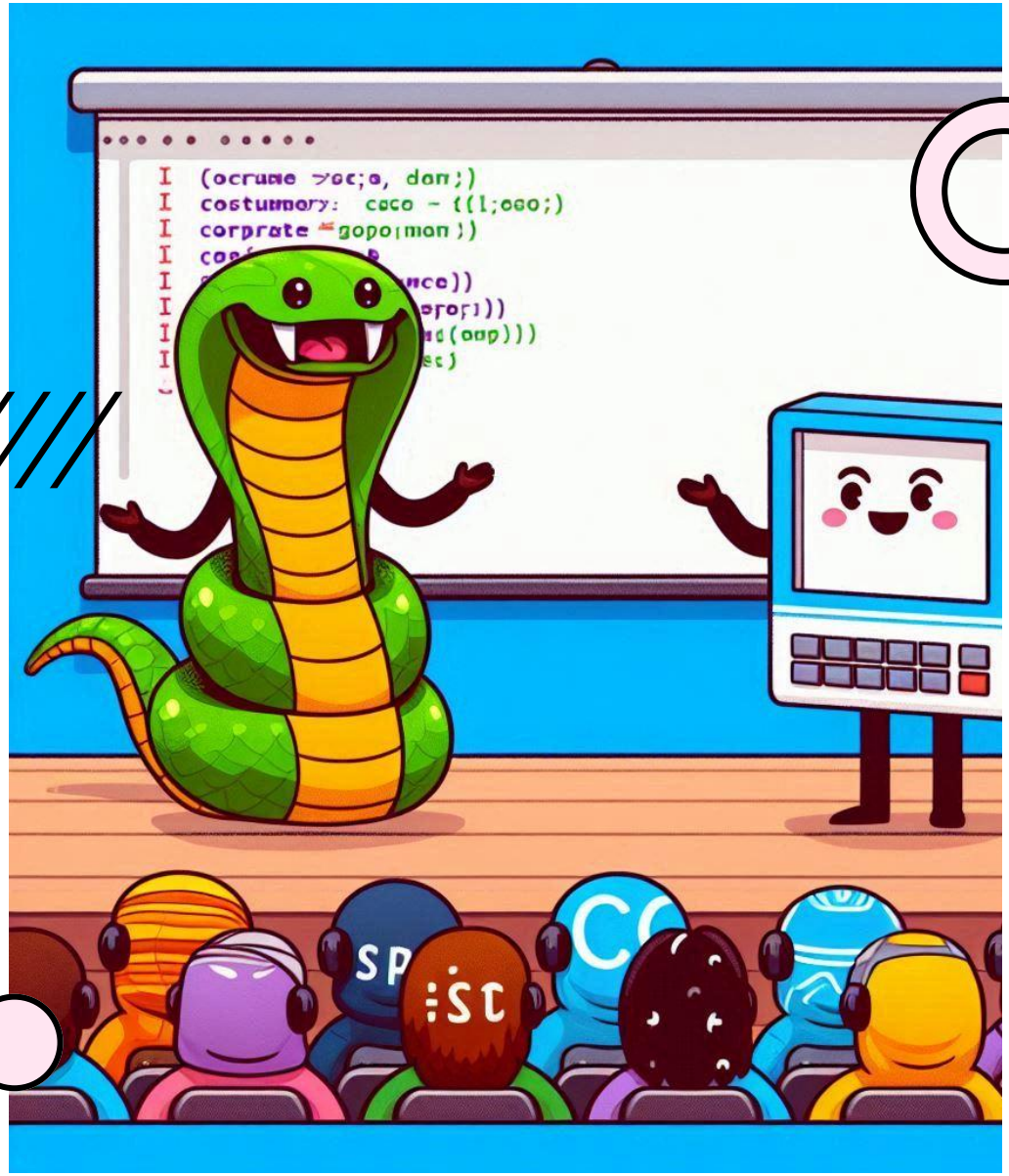
Debugging i C++ (Mac)

- Det finnes en egen oppskrift for dette (se PDF fra forelesningen)

L I V E K O D I N G :

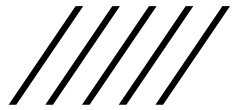
I N T R O
T I L
C + +

"$LastExitCode" i Powershell (Windows) "echo $?" i bash (Mac/Linux)



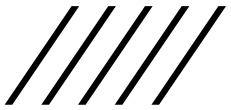
Hvorfor brukes C fortsatt når C++ kan gjøre mer (bl.a. med objekter)?

- C er noe raskere fordi det ikke er *bygd med tanke på* klasser og objekter, blant annet (betyr dog ikke at objekter er *umulig*)
- C er enda mer lavnivå (nærmere maskinvaren) enn C++: "Writing C is just about as close to writing assembly as you can get without writing assembly"
- C er et relativt enkelt språk (kan gjøre relativt få ting, men gjør dem bra)
- C++ er relativt komplisert (kan gjøre relativt mange ting, men kanskje ikke alltid like bra)
- TLDR: Forskjellige språk er gode i forskjellige situasjoner



Arrays vs lister

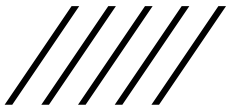
Array	Lenket liste
Kun én datatype	Blande flere datatyper mulig
Fast størrelse	Vokser og krymper etter behov
Elementvise operasjoner (numpy array i Python)	Må bruke løkke for å gjøre noe med alle elementene
Rask og spesialisert	Treig og generell
Bruker indekser (direkte tilgang til elementer)	Må alltid starte først/sist (Python har indeksering)



- Python-lister er strengt tatt en mellomting siden de har indeksering

Litt om bits og bytes

- En bit (*binary digit*) er et siffer i totallsystemet: 0 eller 1
- En *byte* er en etterlevning fra tiden da datamaskiner behandlet 8 bits av gangen (1 byte = 8 bits)
- 01001000 og 01101001 er eksempler på bytes i minnet
  - Kan tolkes som bokstavene “H” og “i” (char)
  - Eller heltallene 72 og 105 (int)
- Nå behandler maskinene *vanligvis* 64 bits av gangen (8 bytes) men byte som enhet har overlevd av historiske årsaker



LIVEKODING:

ARRAYS
I
C++

- Python har mer sikkerhetsnett enn C++
  (C++ går ofte rett på minnet i maskinen)

Hva er en pointer (peker)?

- En variabel er et navn til en verdi som ligger i minnet til maskinen
- En peker er et navn vi gir til *minneadressen* til en slik verdi
- **int e = 10** lagrer verdien 10 på minneadresse 0xA45C tilfeldig adresse som varierer fra gang til gang)
- **&e** gir oss denne adressen (0xA45C) – tilsvarer **id(e)** i Python
- En peker **int* e_sin_adresse = &e** er en variabel hvor verdien er *minneadressen til en annen variabel* (målt i bytes)
- (og pekeren har også sin egen adresse, f.eks. 0xA3C2)

Satt i system:

Type variabel	Definisjon	Minneadresse	En verdi
Vanlig	double pi = 3.14	&pi	pi
Peker	double* pip = &pi	pip	*pip
Array	double pia[] = {pi}	pia (1. element)	*pia (1. element) *eller* pia[0] (hvilket som helst element)

Noen viktige spørsmål

I know this is a really basic question, but I've just started with some basic C++ programming after coding a few projects with high-level languages.

Basically I have three questions:

1. Why use pointers over normal variables?
2. When and where should I use pointers?
3. How do you use pointers with arrays?

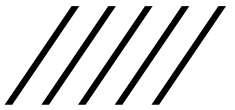
- When and where should I use pointers?

1. Pointers allow you to refer to the same space in memory from multiple locations. This means that you can update memory in one location and the change can be seen from another location in your program.

2. You should use pointers any place where you need to obtain and pass around the address to a specific spot in memory. You can also use pointers to navigate arrays:

- Why use pointers over normal variables?

Short answer is: Don't. ;-) Pointers are to be used where you can't use anything else. It is either because the lack of appropriate functionality, missing data types or for pure perfomance. More below...



LIVEKODING:

PEKERE OG ARRAYS

Etter forelesningen

- Husk fristen for prosjekt 1 i kveld kl. 23:59
  - Dere må kontakte studieinfo@ifi.uio.no ved eventuelle utsettelser
- Hvis du vil bytte gruppe (eller levere alene) på prosjekt 2 og 3, send inn dette nettskjemaet senest **søndag 28/9** kl. 23:59
  - Hvis du er fornøyd med gruppen i prosjekt 1, trenger du ikke gjøre noe
  - Fyll likevel ut hvis andre i gruppen er avmeldt emnet, eller du ikke fikk kontakt med dem