

# Obligatorisk oppgave nr.2

## PROG1003 – Objekt-orientert programmering, våren 2025

**Innleveringsfrist: 25.februar 2025 kl.11:00 (må overholdes)**  
**i Blackboard på pdf-format**

Lag et program som oppretter objekter for *fire* ulike typer utlånsklasser i et bibliotek, og som leser inn noen få data om hvert av dem. Programmet *skal* inneholde og utføre følgende:

- Lag baseklassen `Utlaan` med `string` navn.  
Fra denne avledes og lages `Bok` med `int` ISBN, og `Film` med `int` spilletid.  
Fra `Bok` avledes/lages klassene `Skjonnliteratur` og `Fagtekst`.  
Fra `Film` avledes/lages klassene `Spillefilm` og `Dokumentar`.  
Finn selv på datamedlemmer (ett holder) i de fire siste avledede klassene.  
*Det er disse fire siste vi i praksis skal lage objekter av i programmet vårt.*
- *Alle* data (i de totalt syv klassene) *skal* være `private` – ikke `protected`.  
Dette medfører bl.a. at det *kun* er hvert enkelt objekt som selv leser og skriver *alle* sine egne datamedlemmer, vha. funksjonene: `lesData()` og `skrivData()`.  
`skrivData()` kaller den tilsvarende funksjonen i vedkommendes baseklasse.  
Det gjør *ikke* `lesData()` (se neste punkt).
- Lag en parameterløs constructor inni *hver* av de syv klassene.  
Denne constructoren kaller automatisk vedkommende objekt sin `lesData()`.
- Lag *i tillegg* en constructor i klassene `Fagtekst`, `Bok` og `Utlaan` som tar `navn` som parameter. De to førstnevnte leser objektets datamedlemmer (også ved å kalle `lesData()`), samtidig som de sender `navn` med som parameter oppover til baseklassen. `Utlaan` slipper dermed å lese inn navnet, men kopierer det bare inn i sitt `navn`.  
**Hint:** Vent gjerne med å implementere dette punktet til *alt annet er ferdig og virker* (mer i et annet punkt nedenfor).

- Lag fire globale vectorer som hver inneholder pekere til objekter av de fire klassene.
- Lag et hovedprogram der brukeren får spørsmål om hvilket av de fire objektene som ønskes. For hver:
  - opprettes aktuelt objekt
  - alle data i arvehierarkiet leses inn (constructorene kaller jo klassens `lesData()`)
  - alle objektets data (også de oppover i arvehierarkiet) skrives ut på skjermen
  - peker til objektet legges inn i aktuell `vector`

Dersom brukeren velger å lage en ny `Fagtekst`, skal programmet spørre etter et navn.

Dersom brukeren *kun* svarer ENTER (tom/blank tekst), blir objektet laget *uten* parameter, ellers lages det ved å sende med dette navnet til den andre constructoren (som så igjen sender dette med oppover i arvehierarkiet).

- Det skal også være et menyvalg for å få skrevet ut *alt* i *alle* vectorene.
- Når programmet avsluttes skal *alle* tilpekte objekter i vectorene slettes, og selve vectorene skal nullstilles/cleares.

**NB:** Husk å skrive *ren C++-kode* (og ikke C, med bl.a. `printf`, `scanf`, `gets`, `malloc` og `free`), god og *korrekt* kommentering, innrykk, leselighet og at linjene ikke blir for lange (så teksten wrapper ved en evt. utskrift) - ellers risikerer den dessverre å ikke bli godkjent.

**Lykke til!**

**Malin Foss, Martin Kristensen Eide og FrodeH**