# Opgaver til computationel tænkning

Herunder finder du en række praktiske opgaver/øvelser, som kan være med til at øge forståelsen af computationel tænkning.

## Øvelse 1 - Dekomposition

Eleverne er sammen to og to.

1. Den ene fortæller, hvad han har lavet i tidsrummet fra han stod op i morges til han kom i skole. Den anden skriver forløbet ned (på punktform) – så han kan genfortælle det.
2. En del af forløbet er formentlig at personen har børstet tænder. Den anden elev skal nu udførligt og detaljeret beskrive, hvorledes han har børstet tænder. Den første skriver ned, så han kan genfortælle detaljerne i processen.

Efter opgaven forklarer læreren hvorledes vi nu har beskrevet en proces ved at nedbryde den del processer (stå op, gå på toilet, tage tøj på, spise morgenmad, børste tænder o.s.v.). Han forklarer også, hvorledes man kan dykke ned i en delproces (tandbørstning), og beskrive denne meget mere detaljeret, uden at man mister overblikket.

## Øvelse 2 – Abstraktion

For at forstå begrebet abstraktion skal eleverne lege praktiker/filosof. Dette foregår således:

Eleverne skal stå op.

Den ene elev (praktikeren) nævner en ting (eksempel: blyant). Den anden elev (filosoffen) skal så finde et mere abstrakt begreb for denne ting (eksempel: skriveredskab).

Der findes så mange som muligt på 1 minut (tag tid)

Derefter er det Filosoffens tur til at nævne ting/begrebet (eksempel: Tid). Praktikeren skal så finde en konkret ting som passer til begrebet (eksempel: Ur).

Der findes så mange som muligt på 1 minut (tag tid)

Det forklares nu hvorfor abstraktion er vigtigt i forbindelse med softwareudvikling.

1. Fordi man gennem abstraktion kan bestemme detaljeringsgraden, og abstrahere fra ting der allerede er lavet. (da vi talte om tandbørstning i førte omgang valgte vi formentlig at abstrahere fra detaljerne)
2. Fordi man gennem abstraktion kan gøre løsninger generelle. (At spise morgenmad kan være den samme proces uanset hvad vi spiser. Vi forbereder maden, serverer den, spiser den og rydder væk)

## Øvelse 3 – Mønstergenkendelse

Øvelsen laves i plenum. Ønske mere avancerede mønster øvelser, som eleverne selv kan gennemføre, henvises der til diverse IQ test, som ofte baseres på logisk tænkning.

Fortsæt rækkerne

1. 1,2,3,4, ?  
   (Svar: 5 – fortsat talrække)
2. 1, 4, 9, 16, 25, ?  
   (Svar 36 – kvadrat tal)
3. 1,2,3,5,7,11,13,?  
   (Svar: 17 – primtal)
4. X,O,X  
   O,X,O  
   X,O,?  
   (Svar: X)
5. O,X,=,O  
   X,=,O,?  
   (Svar: O)
6. LA,EB,FC  
   FC,LA,EB  
   EB,FC,??  
   (Svar: LA)

I softwareudvikling er det vigtigt at kunne se og identificere gentagelser og mønstre. Derved kan vi lave algoritmer, som kan løse opgaverne. Computeren er godt til at lave gentagelser (løkker), så derfor er det smart at lave løsninger baseret på brug af gentagelser.

## Øvelse 4 - Algoritme

Eleverne skal nu i grupper på 2 personer lave en opskrift for brygning af kaffe ved brug af en kande og en kaffetragt med filter. Opskriften skal være så detaljeret og præcis som muligt.

Løsningen kunne se således ud:

1. Vandet koges
2. Kanden tømmes for rester og skylles med en smule kogende vand
3. Tragten sættes på kanden
4. Filter puttes i tragten
5. Der puttes kaffe i filteret (1 strøget måleske kaffe pr 2 kopper. Skal den være stærk doseres 1 måleske med top pr 2 kopper)
6. Der hældes kogende (94 grader) vand på op til 1 cm under filterkant
7. Når vandet er løbet igennem, fyldes der på igen, og dette gentages indtil du har den ønskede mængde kaffe.
8. Filteret med kaffegrums kasseres (til kompost), tragten tages af, og der sættes låg på kanden.

Denne opskrift kan også kaldes en algoritme. En algoritme beskriver hvorledes noget laves enten ved hjælp af en proces beskrivelse eller ved brug af formler. I dette tilfælde gør vi brug af begge dele.

Øvelsen kan laves på andre dagligdags processer, som eleverne kender i forvejen. Procesbeskrivelser kan dokumenteres i flowdiagrammer.