# Multiple Choice and Repetition

|  |  |
| --- | --- |
| **Læringsmål** | Alle tidligere læringsmål |
| **Forventet læringsudbytte [SOLO]** | Alle tidligere læringsudbytter |
| **Din forberedelse** | * Overvej, hvilke emner du har været igennem hen over de første ca. 3 uger på studiet * Bemærk der er ingen nye fagemner til i dag, kun repetition af tidligere emner |

**OBS: I denne undervisningsgang er der ingen undervisere til stede (vi er på kursus).**

Snart (om en uge) afholder vi en multiple choice test. Den skal du være med til at udforme. Derfor er det vigtigt, at du har overvejet, hvilke emner du har været igennem hen over de første ca. 3 uger på studiet.

Opgaven er derudover tilpasset således, at du har mulighed for at gense tidligere opgaver og løse dem igen, eller færdiggøre dine løsninger. Gør brug af hinanden i teamet/holdet/årgangen, hvis I har problemer og gerne vil sparre med nogen.

Du vælger selv, om du vil arbejde online eller nede i boksen. Det er frivilligt. Aftal det meget gerne i dit team, hvordan I gør det.

# Dagens quote:

[Alan Turing](https://en.wikipedia.org/wiki/Alan_Turing): *‘We can only see a short distance ahead, but we can see plenty there that needs to be done.’*

# Øvelse 1: Mind-map øvelse

Snart (inden uge 42) afholder vi en multiple choice test. Den skal du være med til at udforme.

Her er der en liste af kategorier og emner, som der skal benyttes i nedenstående opgave:

* Computational Thinking
* C#-datarepræsentation – variabel, erklæring, virkefelt
* C#-datatyper – int, string, char, bool, double, array
* C#-datahåndtering – tildeling, udtryk, initialisering, typekonvertering og casting
* Programflow – forgrening (if-else, switch), løkke (while, do-while, m.m.), metodekald/-retur, exceptions
* C#-klasse – klassedefinition, medlemmer (felt, metode, constructor, property), access modifier, access keywords
* OOP-principper (Indkapsling)
* Versionsstyring (Git)
* Debugging
* Unit testing

**Fremgangsmåde:**

1. Se på listen af udleverede emner, som du har været igennem hen over de første ca. 3 uger
2. Find på et spørgsmål til emnet, der kan belyse, at man har viden om det
3. Beskriv det korrekte svar, og find på **to** forkerte svarmuligheder
   * Selvom det er forkerte svarmuligheder, skal de stadig være relateret til emnet
   * Skriv ”Korrekt svar” og ”Forkert svar” ud for hver svarmulighed
4. Del spørgsmålet og svaret med din sidemand, begrund
5. Del det med hele bordet
   * Dokumentér dette i et fælles dokument
6. Gå tilbage til punkt 1

*Tidsramme: 45 minutter (Hold selv styr på tiden!)*

Alle bliver enige om de bedste spørgsmål og besvarelser (som I har i jeres fælles dokument) og sender en mail med dette til følgende (begge) e-mails:

* [jbro@ucl.dk](mailto:jbro@ucl.dk)
* [ldki@ucl.dk](mailto:ldki@ucl.dk)

# Øvelse 2: Læringsudbytteplan (individuel)

I forbindelse med alle emner som er gennemgået i tidligere opgaver i forberedelsen, start på eller opdatér din læringsudbytteplan (se dagens ord i Ex02-Conditionals samt slide 19 og 20 i Ex02-Slides). Dvs. vurdér og begrund dit SOLO-niveau på hvert emne samt anfør dine tiltag.

# Øvelse 3: Konsolidering

Dagen er tiltænkt som en konsolideringsdag. Det vil sige, at du ikke vil blive introduceret til noget nyt på denne dag, men du kan vælge selv, hvordan du vil bruge dagen, hvor du har brug for det:

1. Gå tilbage til de øvelser i tidligere opgaver, du ikke har fået løst endnu, og gøre dem færdig; du vælger selv.
2. Genløs alle programmeringsøvelser i tidligere opgaver med den indsigt, du har nu.
3. Hvis du har løst alle opgaver, så gennemgå og revurdér dem og se, om der kunne være behov for at tilpasse koden; f.eks. gøre den mere effektiv, kortere, mere læselig eller bedre til at håndtere fejlsituationer, mm.
4. Eller arbejd på det projekt, du har behov for lige nu (f.eks. Hydac)

Anvend densamarbejdsform, der passer bedst til din aktivitet individuelt eller i samarbejde med dit team.

God fornøjelse.

# Øvelse 4: (Bonus) Hva' er du for'n en?

Kun udfør følgende del-øvelser, hvis du og dit team har tid.

Du skal i denne øvelse undersøge om Tigerdyret (en: Tigger) og Peter Plys (en: Winnie the pooh) når at møde hinanden inde i 100-meterskoven.

## Øvelse 4.1: Klargøring

Forudsætningen for at følge den følgende øvelse er, at du har oprettet en konsolapplikation, tilføjet et unit test projekt og har kopieret unittestene fra ’Ex13-MultipleChoiceAndRepetition.UnitTest.docs’ ind i unit test projektet. Disse unittests befinder sig i ’Supplerende materiale’ på Git (Benyt git pull).

## Øvelse 4.2: Beskrivelse af opgaven

Benyt parprogrammering.

Tigerdyret og Peter Plys starter ved hver sin position i 100-meterskoven, og de bevæger sig med hver sin hastighed. Deres startplacering er beskrevet ved et tal, der repræsenterer deres position på x-asken. Deres bevægelse er også beskrevet med et tal, der repræsenterer, hvor meget de rykker hen ad x-aksen per runde.

Du skal udforme en statisk metode, der skal acceptere 4 tal:

Tal 1: Skal beskrive startplaceringen for Tigerdyret på x-aksen

Tal 2: Skal beskrive, hvor langt Tigerdyret hopper hen ad x-aksen per runde

Tal 3: Skal beskrive startplaceringen for Peter Plys på x-aksen

Tal 4: Skal beskrive, hvor langt Peter Plys går hen ad x-aksen per runde

Den statiske metode skal returnere en tekst-streng, der skal indikere, om Tigerdyret og Peter Plys er lykkedes med at mødes eller ej.

Hvis det er lykkedes, skal x-koordinaterne for, hvor Tigerdyret og Peter Plys står på mødestedet, returneres, for eksempel: 12,12

Hvis ikke det lykkedes, skal der returneres beskeden: NO

Du skal kun tage højde for tal mellem 0 og 10000 (*Vi antager, at skoven i virkeligheden er 10000 meter lang*).

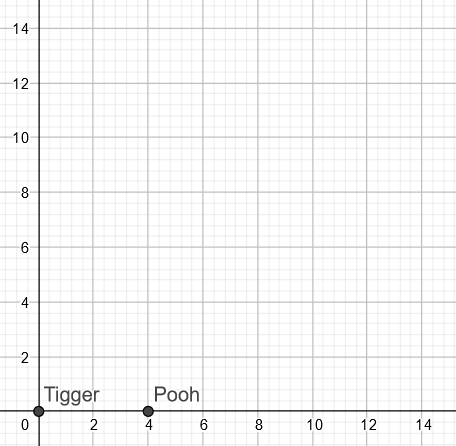
Undervejs i din udformning af din løsning, skal du benytte dig af debugging.

**Lad os se på et konkret eksempel**:

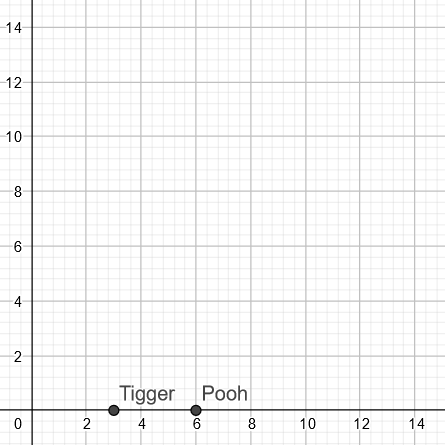
Vi modtager følgende input: 0,3,4,2

Det betyder, at Tigerdyret starter på x-aksen ved placeringen 0. Og han bevæger sig hen ad x-aksen med 3 per runde. Imens starter Peter Plys på x-aksen ved placeringen 4. Og han bevæger sig hen ad x-aksen med 2 per runde. Gennemgang af dette forløb:

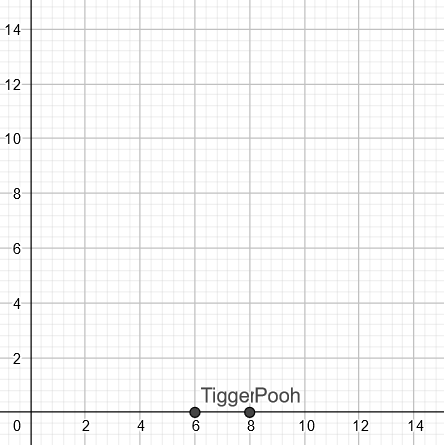
* Start position: 0, 4



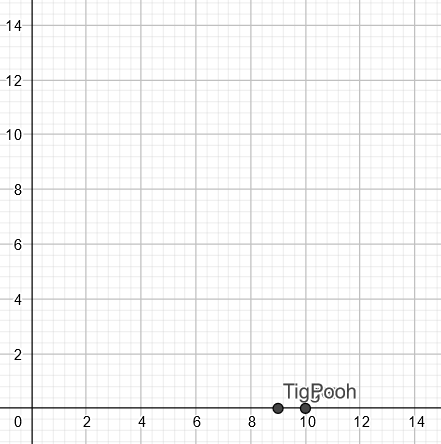
* Runde 1: 3, 6



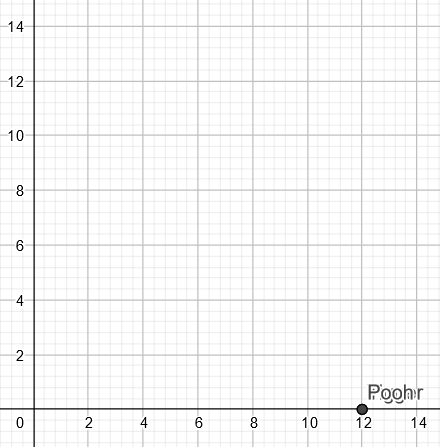
* Runde 2: 6, 8



* Runde 3: 9, 10



* Runde 4: 12, 12



Det vil sige, at de støder på hinanden efter 4 runder. Ergo er dette input sandt.