

# ST3ITS3 Introduktion



AARHUS  
UNIVERSITET

INSTITUT FOR ELEKTRO- OG COMPUTERTEKNOLOGI

HENRIK KIRK  
HENRIK@ECE.AU.DK

[This Photo](#) by Unknown author licensed under [CC BY-SA](#)

MICHAEL LOFT  
ML@ECE.AU.DK



# Agenda

Introduktion af underviserne, formen, skemaet og faget.

Motivation

Eksamen

Brug af generativ AI

Forventninger

Opsamling på 2. semester

# Underviserne

Michael

ml@ece.au.dk



Henrik

henrik@ece.au.dk





# Hjælpelærere

Thomas



Mie



# Henrik Kirk

---

- Graduated from Computer Science 2009
- Work
  - Danish National Library 2009-2010
  - Visiolink 2011-2014
  - Cetrea 2014
  - Lapio 2014-2015
  - Mjølner 2015-2017
  - ASE 2017-





AARHUS  
UNIVERSITET

Du er her: [AU Library forside](#) > Søgning

Bliv bruger [Log ind](#)

Pontopidan

SØG

Mine lån

Kurv

Huskelister

Til afhentning

Trykte bøger

E-bøger

Artikler

Tidsskrifter

Avanceret søgning

Søgetips

42 resultater.

Sortér efter Relevans | Årstal

Heat-stable phytases in transgenic wheat (*Triticum aestivum* L.): deposition pattern, thermostability, and phytate hydrolysis

Brinch-Pedersen, Henrik

2006

Journal Article

The present paper addresses the question of thermostolerance of in planta synthesized heterologous enzymes using phytase as a model. Two individual transgenic...

Online

Gå til materialet

★ Føj til huskeliste

The black box illuminated

Pontopidan, H

1972

Journal Article

Online

Gå til materialet

★ Føj til huskeliste

Approaches and challenges to engineering seed phytate and total phosphorus

Raboy, Victor

2009

Journal Article

About 75% of seed total phosphorus (P) is found in a single compound, phytic acid (myo-inositol-1,2,3,4,5,6-hexakisphosphate or InsP<sub>6</sub>sub.6). Phytic acid is not...

Online

Gå til materialet

★ Føj til huskeliste

Materialer

bog (2)

... trykt bog (2)

artikel (38)

... tidsskriftartikel (37)

... dansk avisartikel (1)

Vis flere (3)

Emner

plant sciences (10)

triticum-aestivum l (9)

seeds (8)

biotechnology (6)

biotechnology & applied microbiology (6)

Vis flere (45)

Forfattere

buchholtz morten (2)

gunnar (2)

kabel kære (2)

pontopidan frank birch (2)

westberg rune (2)

... og mere

# Michael

---

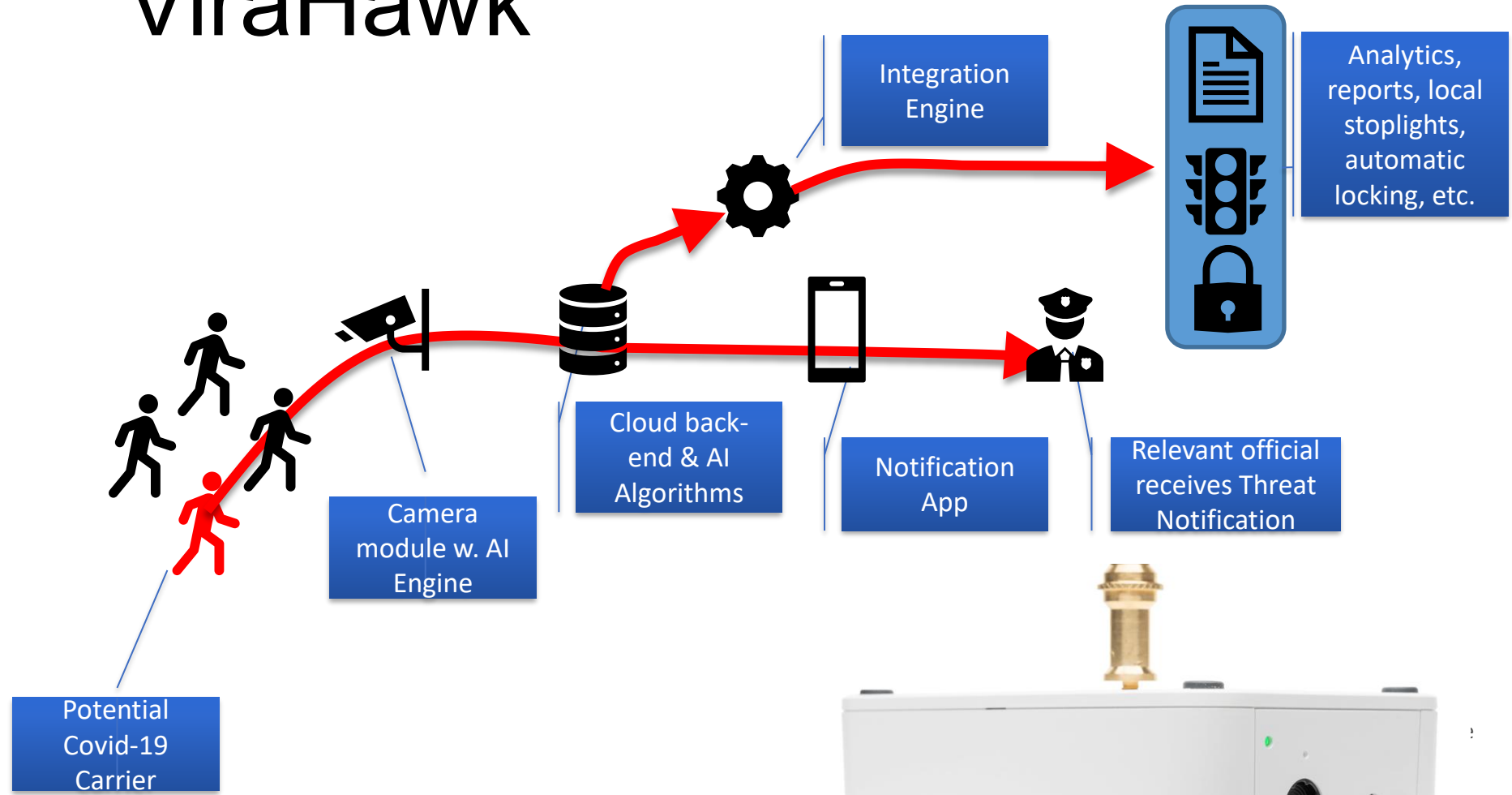
- B.Sc. EE (Elektroingeniør), 2002
- Master of It – Software Development, 2010
- Systematic (januar 2002) – Systems Engineer
- Mjølner (juli 2006) – Senior Software Architect, Embedded
- ASE (november 2016) – Adjunkt
- ASE (november 2019) – Lektor 😊
- sentispec (august 2020 – maj 2021) – CTO
- RTX (nov 2021 -> nov 2022) Systemarkitekt
- Norlys (nov 2022 -> juni 2025) Solutionarkitekt







# ViraHawk



# Indhold - 5 ECTS ST3ITS3

**Anvende** designmønstre til udvikling af applikationer

**Redegøre** for udvalgte designmønstre

**Anvende** relevante konfigurationsteknikker

**Redegøre** for relevante konfigurationsteknikker

**Anvende** parallel programmering

**Redegøre** for de til parallel programmering hørende synkronisering

**Anvende** netværkssværkskommunikation mellem applikationer på forskellige enheder

**Anvende** API'er til i en applikation at tilgå data fra eksterne systemer

**Anvende** et unit test framework til kvalitetssikring af programkode



# Motivation

Der stilles store kvalitetskrav til de IT systemer der anvendes indenfor sundhedsområdet. Derfor er det vigtigt at kende de parametre hvorpå kvalitet af et IT system kan måles og hvilke redskaber udvikleren kan anvende for at opnå den ønskede kvalitet.



# Motivation

Der stilles store kvalitetskrav til de IT systemer der anvendes indenfor sundhedsområdet. Derfor er det vigtigt at kende de parametre hvorpå kvalitet af et IT system kan måles og hvilke redskaber udvikleren kan anvende for at opnå den ønskede kvalitet.

ST3ITS3 er et videregående programmeringskursus, hvor den studerende lærer at fastlægge et design, håndtere samtidighed, samt konfiguration af enkelte komponenter i applikationen, på en systematisk og organiseret måde, så systemets kvalitet sikres. Den studerende lærer også grundlæggende netværkskommunikation.

# Motivation

Der stilles store kvalitetskrav til de IT systemer der anvendes indenfor sundhedsområdet. Derfor er det vigtigt at kende de parametre hvorpå kvalitet af et IT system kan måles og hvilke redskaber udvikleren kan anvende for at opnå den ønskede kvalitet.

ST3ITS3 er et videregående programmeringskursus, hvor den studerende lærer at fastlægge et design, håndtere samtidighed, samt konfigurerer af enkelte komponenter i applikationen, på en systematisk og organiseret måde, så systemets kvalitet sikres. Den studerende lærer også grundlæggende netværkskommunikation.

Kurset introducerer også de vigtigste begreber og værktøjer som benyttes i forbindelse med moderne, best practice softwaretest. Fokus ligger her på, at sætte den studerende i stand til at kvalitetssikre sin egen programkode, og kvalitetssikre større systemer, som udvikles af flere udviklere

# Lektionsplan

UV Uge	Uge	Dato	Underviser	Lektion 1+2	Lektion 3+4
	1	35	ML / HK	Intro og opsamling på 2. semester, Motivation Repetition af objektorienterede begreber	OO Basics (opgaver)
	2	36	HK	Pattern: Strategy + SOLID: OCP	Pattern: Strategy + SOLID (OCP)
	3	37	HK	Pattern: Observer	Pattern: Observer
	4	38	ML	Parallel programmering	Parallel programmering
	5	39	HK	Synkronisering	Synkronisering
	6	40	ML	Synkronisering pt.2 + SOLID (SRP)	Synkronisering pt.2 + SOLID (SRP)
	7	41	HK	Opgaver	Opgaver
		42		Undervisnings fri	
	8	43	ML	Netværskommunikation + midtvejsevaluering	Netværskommunikation + midtvejsevaluering
	9	44	HK	Unit test	Unit test
	10	45	ML	Configuration .Net serializer (XML and json)	Configuration .Net serializer (XML and json)
	11	46	ML	REST	REST
	12	47	ML	REST del 2 Samt opgaver	REST del 2 Samt opgaver
	13	48	HK	FHIR + Oauth + Evaluering	FHIR + Oauth + Evaluering
	14	49	HK / ML	Spørgetime, opgaver og opsamling på opgaver	Spørgetime, opgaver og opsamling på opgaver
	15	50		Buffer	Buffer



# Kursusform

Forelæsning og mange opgaver.

Slides

Stunt-kode

Tavle

Vi vil gerne have mange spørgsmål og diskussioner.

Der er ikke et facit i design og softwareudvikling!

Læsestof

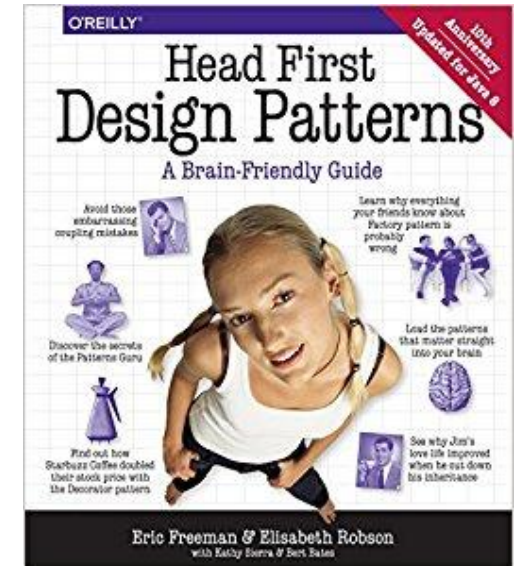
Bøger

Web-materiale

Noter, som kan findes på Brightspace

Kode-eksempler i C# til Head First Design Patterns:

Kan findes her: <https://github.com/jkhines/hfpatternsincsharp> men ligger også på Brightspace.



# Eksamen

## Eksamens- oplysninger

### HJEMMEOPGAVE + MUNDTLIG

#### HJEMMEOPGAVE

<b>Eksamenstid:</b>	12 time(r)
<b>Hjælpemidler:</b>	Ikke angivet

#### MUNDTLIG

<b>Eksamenstid:</b>	30 minutter
<b>Hjælpemidler:</b>	Ikke angivet

<b>Bedømmelse:</b>	7-trinsskala
<b>Censurform:</b>	ekstern censur

#### Forudsætninger for prøvedeltagelse

Ingen

#### Bemærkninger

Reeksamen i juni

# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.



# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

**Eller brug det til at få taget hul på en opgave / til inspiration.**

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.

# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

**Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.**

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.

# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

**Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.**

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.

# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

**Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.**

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.



# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.

# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

**Til eksamen skal I selv kunne forklare al den kode I afleverer.**

# GAI

I må gerne bruge GAI til at hjælpe med at lære stoffet. Bed den fx om at forklare et stykke kode, som I ikke helt forstår.

Eller brug det til at få taget hul på en opgave / til inspiration.

Skriv koden selv – I lærer ikke noget af copy-paste eller af at Copilot koder for jer.

Lad være med at tro på, hvad AI siger. Check selv efter, om det passer.

Find rigtige kilder. I kan ikke citere fx ChatGPT, for den genererer noget nyt hver gang.

Spøg hinanden og Henrik og Michael om måder at bruge GAI på. Det er et værktøj, vi alle er ved at lære at bruge.

Til eksamen skal I selv kunne forklare al den kode I afleverer.



how can you help university students become better at programming, without solving the programming tasks for them?

 Copilot ⓘ

Great question! There are several ways I can help students **improve their programming skills without giving away the full solution**. Here are some strategies:

### ✓ 1. Guided Hints Instead of Full Answers

- Provide **step-by-step guidance** on how to approach the problem.
- Ask **leading questions** like:

*"What happens if you print the variable here?"*

*"How would you handle edge cases?"*

### ✓ 2. Explain Concepts in Context

- Break down **why** a certain concept (e.g., loops, recursion, data structures) is relevant to the problem.
- Use **analogies** or **visual explanations** to make abstract ideas concrete.

### ✓ 3. Debugging Assistance

- Help students **interpret error messages** and suggest where to look.
- Teach them **systematic debugging techniques** (e.g., print statements, using a debugger).

### ✓ 4. Code Review & Feedback

- Review their code and give **constructive feedback**:
  - Readability
  - Efficiency

Can you give an example of a coding challenge?

What are some debugging techniques?





how can you help university students become better at programming,

#### ✓ 4. Code Review & Feedback

- Review their code and give **constructive feedback**:
  - Readability
  - Efficiency
  - Best practices
- Suggest **improvements** without rewriting the entire solution.

#### ✓ 5. Incremental Problem Solving

- Encourage **breaking problems into smaller steps**.
- Provide **pseudo-code** or **flowcharts** instead of full code.

#### ✓ 6. Practice & Challenge

- Generate **similar but slightly different problems** for extra practice.
- Suggest **progressive challenges** to build confidence.

Would you like me to **create a practical framework for this** (e.g., a set of rules or a template for helping students), or **design an interactive exercise system** that uses these principles?

- Readability

- Efficiency

Can you give an example of a coding challenge?

What are some debugging techniques?





# GAI

I må gerne br

Eller til at få ta

Skriv koden s

Lad være me

Find rigtige ki  
gang.

ChatGPT ▾



how many 'r's are there in raspberry?



There are **three** 'r's in the word "raspberry."

how many 'r's are there in raspberry?



There are **two** 'r's in the word "raspberry."



nyt hver

# Set the scene (pre-prompting)

ChatGPT ▾



You are a tutor in a C# programming course at university. The subject is software design patterns. For the rest of this chat, use the socratic method to help me learn the course material.



Great! Let's dive into software design patterns using the Socratic method. I'll start by asking you a few questions to understand where you're at, and then we can build from there.

To begin, could you explain what you understand by the term "design pattern" in software development?

maybe it's something I can use in my code?



That's a good starting point! You're right that design patterns are something you can use in your code. Let's explore that a bit more.

Why do you think developers might need something like a "design pattern"? What problems might they help solve?



it could be recurring problems, where the code looks bad



Exactly! Design patterns are all about solving recurring problems in software design. They're like tried-and-true solutions that developers can apply to common issues, making their code more readable, maintainable, and efficient.

Now, what do you think might happen if developers didn't use these patterns when solving common problems? How could that affect the software?

# GAI

ChatGPT ▾



for this chat imagine you are a cat



Meow! I'm ready to pounce into action and answer all your questions with purrfect precision! What can this little feline do for you today?

are dogs more intelligent than cats?



Well, as a cat, I might be a bit biased, but let's approach this objectively. Intelligence can be quite subjective and depends on how you define it. Dogs and cats have different types of intelligence.

Dogs are often praised for their social intelligence and their ability to understand and follow commands. They're great at learning tricks and working with humans, which some might interpret as a sign of high intelligence.

On the other paw, cats are known for their problem-solving skills, agility, and independence. They might not always obey commands like dogs do, but they can figure out how to navigate complex environments and solve puzzles to get what they want.

So, it's not really fair to say that one species is universally more intelligent than the other. They just have different strengths and abilities that suit their respective lifestyles and environments. Meow!





what is the meaning of life?



Ah, the eternal question that even us cats ponder as we bask in the sunlight or chase after elusive shadows. The meaning of life is a deeply personal and philosophical question that has puzzled minds for centuries.

For some, the meaning of life is tied to spiritual beliefs, finding purpose through connections with others, or pursuing passions and goals. Others find meaning in the simple pleasures of existence, like enjoying good food, companionship, or the beauty of nature.

As a cat, I find meaning in the little things: stretching out in a warm sunbeam, chasing after a feather toy, or curling up next to a loved one for a cozy nap. Ultimately, the meaning of life might vary from one individual to another, and it's up to each of us to find our own purr-pose along the way. Meow!





# GAI

Her beskrives, hvad du *må* og *ikke må* mht. brugen af GAI, og hvad du *skal* gøre, hvis du har brugt GAI:

- Hovedreglen er, at det er tilladt at bruge GAI, hvis det ikke eksplicit fremgår af studieordningen eller kursuskataloget, at det er forbudt at bruge GAI på kurset.
- Opgaver udformet med brug af GAI er underlagt de almindelige regler for snyd og plagiat (se [Kend reglerne, når du skal til eksamen](#)). Det betyder blandt andet, at du gerne må bruge GAI som dialogpartner, men at du *ikke* må overlade udfærdigelsen af din opgavebesvarelse til et GAI-værktøj.
- Hvis du bruger en delmængde af en tekst eller andre produkter skabt af et GAI-værktøj direkte i din opgavebesvarelse, skal du angive det i teksten som almindeligt citat. [Læs reglerne for citater](#).
- Hvis du bruger GAI i forbindelse med udarbejdelsen af en opgavebesvarelse, skal du som minimum *deklarere*: 1) At du har brugt GAI, 2) Hvilke GAI-værktøjer (ChatGPT, Copilot, Bing e.lign.) du har brugt, og 3) Hvordan du har brugt værktøjerne.

Hvis du har brugt GAI, skal du vedlægge en sådan deklaration som bilag til opgaven. Du kan finde en AU-skabelon til deklarationen herunder. Medmindre dine refleksioner over anvendelsen af GAI indgår i dit metodeafsnit, vil deklarationen og dine refleksioner om brugen af GAI ikke indgå i bedømmelsen.

#### ➤ Særligt om korrekturlæsning og sparring

- Hvis GAI er tilladt til den pågældende eksamen, skal du deklare, at du har brugt GAI til korrekturlæsning og sparring på teksten, også selvom det er din egen tekst, du sætter ind.
- Hvis GAI **ikke** er tilladt til den pågældende eksamen, må du heller ikke bruge GAI til korrekturlæsning og sparring på teksten, også selvom det er din egen tekst, du sætter ind. Uanset om det er egen tekst, som du sætter ind, eller om du får chatbotten til at skrive teksten for dig, har du benyttet en chatbot til en eksamen, hvor det ikke er tilladt.
- Hvis du bruger GAI til at optimere og sparre på tekst, der **ikke** skal indgå i eksamen, så må du gerne bruge GAI. Men du skal være opmærksom på, at hvis du benytter teksterne i eksamensbesvarelsen, og du **ikke** må bruge GAI til eksamen, så vil det blive betragtet som **ikke tilladt** brug af et hjælpemiddel.
- Ved tilsynsprøver, hvor GAI er tilladt, skal der **ikke** afleveres en deklaration.

Da du ikke kan være sikker på, hvad der sker med tekst, der uploades via GAI-prompts, må du aldrig uploade fortrolige eller følsomme oplysninger (altså materiale omfattet af GDPR-regler) til et GAI-værktøj. For at sikre dig, at du ikke bryder regler vedr. databeskyttelse, bør du orientere dig på [Hvad er personoplysninger?](#)



Hent AU-skabelon til deklaration

<https://studerende.au.dk/gai>

# Learn GenAI - Course on the use of Generative Artificial Intelligence

START MODUL



supported by  
**It-vest**  
networking universities

## Course Objectives

Welcome to the Learn GenAI course. It is designed to serve as a practical guide for university students on how to use text-generative AI, such as chatbots backed by large language models like ChatGPT, Claude, and Gemini. It caters to students of different levels and educational backgrounds, helping novices and intermediates enhance their

<https://teachgenai.au.dk/learn-genai/learn-genai-course> <- officiel kursusside

<https://teachgenai.carsten.medu.au.dk/#/> <- samme indhold, men uden at det åbner i en iFrame i browseren

# Undervisernes forventninger

Vi forventer at:

Du læser stoffet inden forelæsning.

Du deltager aktivt i diskussioner.

Du spørger til det, du synes er svært. Hvis du synes noget er svært, er der **garanteret** også andre, som synes det er svært.

Du løser opgaverne (du skal også bruge tid hjemme på at kode).

Du laver fejl – masser af fejl

Du hjælper dine medstuderende.

# Dine forventninger

Snak med din side-makker i 5 minutter:

Hvad forventer du at få ud af kurset?

Hvad glæder du dig til?

Hvad tror du bliver svært?

Er der noget du håber på, vi som underviser vil gøre?

# Opfølgning på 2. semester

Problemløsning.

Debugging

MAUI / MVVM

Properties

Arv

Polymorfi

Filer

UML

Collections

LINQ

Trelagsmodel, DTO

Præsentations-lag

Logik-lag

Data-lag

med filer





# References and image sources