



# UiT Norges arktiske universitet

<b>Kort hjemmeeksamen i:</b>	INF-1400 Objektorientert Programmering
<b>Dato:</b>	2020-09-21 til 2020-09-23
<b>Tidspunkt:</b>	09:00 (21.) til 15:00 (23.)
<b>Kursansvarlig:</b>	John Markus Bjørndalen Edvard Pedersen
<b>Antall sider:</b>	
<b>Support:</b>	Du kan ringe 776 20 880 for support på eksamensdagen.
<b>Vekting av spørsmål, eller annen informasjon:</b>	
<b>Viktig informasjon om sitering og plagiering:</b>	<ol style="list-style-type: none"><li>1. Dette er en individuell eksamen som skal besvares uten samarbeid med andre.</li><li>2. Alle hjelpemidler er tillatt (egne notater, pdf'er fra forelesningene, lærebok, internett etc).</li><li>3. Alle eksamener som leveres i WISEflow blir automatisk sjekket for plagiat. Det er ikke tillatt å kopiere medstudenter, nettressurser, kilder, eller litteratur uten referanser.</li></ol>

## Praktisk informasjon:

- Oppgaven er en forenklet versjon av en obligatorisk oppgave og er ment å trenge 3-4 timer.
- Innleveringsperioden går over tre dager. Lever tidlig for å unngå tekniske problemer på slutten.
- Dere kan levere pseudokode. Det er ikke behov for å lage en kjørende løsning. Vi er ute etter å se mønster og tenking. Prøver dere å lage en kjørende løsning kommer dere raskt til å bruke mer tid enn det vi krever.
- Ikke overdriv kompleksiteten i løsningen (f.eks. modellere ting som ikke er beskrevet i oppgaven). Vi skal ikke lage en komplett løsning på 3 timer. Det betyr at noen ting kan være litt urealistiske. Fokuser på hvordan du skal jobbe med koden og designet på programmet.
- Vi vil være tilgjengelige på telefon, Discord og Teams hvis det er spørsmål under eksamen.
- Det er lov å stille spørsmål hvis det er noe dere ikke forstår i den utleverte koden.

## Bakgrunn

En kjenning av deg fikk akkurat skadet bilen sin etter å ha havnet i et hull i veien. For å undersøke spriket mellom faktisk vedlikehold og behov for vedlikehold for å opprettholde en god veistandard trenger vi en simulering.

I første runde begynner vi med en simulering som viser hvor mye veiene blir slitt. Kjenningen din har lagd et utkast, men det gikk litt fort for seg siden han bruker mye av tiden sin på verkstedet og i telefon med forsikringsselskapet. Vi må gjøre betydelige forbedringer for å kunne jobbe med koden framover.

Ta utgangspunkt i den utleverte koden. Den bruker ikke klasser og objekter, men nå er oppgaven din å endre koden slik at du kan demonstrere dette.

## Oppgave 1)

**Lag et klassediagram for de klassene du tenker er naturlige å ta med, inkludert relasjoner mellom klassene.**

Løsningen skal demonstrere arv og polymorfi. Hvis du mener dette er en klønete løsning kan du gjerne kommentere dette i oppgave 2 og legge til hvordan du heller ville gjort det.

## Oppgave 2)

**Beskriv diagrammet og endringene du gjorde fra det originale designet.**

## Oppgave 3)

**Implementer det du har skissert.**

## Oppgave 4)

Dette er en oppgave som skiller seg litt ut: her kan du selv foreslå hva du ønsker å gjøre for å demonstrere et konsept fra objektorientering.

**Demonstrer spesialisering av en klasse ved å gjøre følgende:**

- a) **Foreslå noe du kan legge til og beskriv hvordan det du legger til demonstrerer spesialisering.**
- b) **Implementer det du har foreslått over.**

## Oppgave 5)

- a) **Hvilket av begrepene has-a og is-a korresponderer til begrepene komposisjon og arv?**
- b) **Hvor i designet bruker du de to begrepene?**
- c) **Hvis du vil legge til en bompengebrikke i en bil: Hvilke av de to begrepene over vil du bruke, og hvordan vil du legge det inn i programmet du har nå? Begrunn valgene dine.**