

i Forside

Institutt for datateknologi og informatikk

Eksamensoppgave i DCST1003 Grunnleggende programmering

Eksamensdato: 14. desember 2020

Eksamenstid (fra-til): 0900-1300 (4 timer)

Hjelpemiddelkode/Tillatte hjelpemidler: A / Alle hjelpemidler tillatt

Faglig kontakt under eksamen:

Tlf.: Atle Nes 98852760, Ole Christian Eidheim 90551635

Teknisk hjelp under eksamen: NTNU Orakel

Tlf: 73 59 16 00

ANNEN INFORMASJON:

Gjør dine egne antagelser og presiser i besvarelsen hvilke forutsetninger du har lagt til grunn i tolkning/avgrensning av oppgaven. Faglig kontaktperson skal kun kontaktes dersom det er direkte feil eller mangler i oppgavesettet.

Lagring: Besvarelsen din i Inspira Assessment lagres automatisk hvert 15. sekund. Jobber du i andre programmer – husk å lagre underveis.

Juks/plagiat: Eksamen skal være et individuelt, selvstendig arbeid. Det er tillatt å bruke hjelpemidler. Under eksamen er det ikke tillatt å kommunisere med andre personer om oppgaven eller å distribuere utkast til svar. Slik kommunikasjon er å anse som juks. Alle besvarelser blir kontrollert for plagiat. [Du kan lese mer om juks og plagiering på eksamen her.](#)

Varslinger: Hvis det oppstår behov for å gi beskjeder til kandidatene underveis i eksamen (f.eks. ved feil i oppgavesettet), vil dette bli gjort via varslinger i Inspira. Et varsel vil dukke opp som en dialogboks på skjermen i Inspira. Du kan finne igjen varselet ved å klikke på bjella øverst i høyre hjørne på skjermen. Det vil i tillegg bli sendt SMS til alle kandidater for å sikre at ingen går glipp av viktig informasjon. Ha mobiltelefonen din tilgjengelig.

Vekting av oppgavene: Hvordan oppgavene er vektet framgår i oppgavesettet.

OM LEVERING:

Besvarelsen din leveres automatisk når eksamenstida er ute og prøven stenger, forutsatt at minst én oppgave er besvart. Dette skjer selv om du ikke har klikket «Lever og gå tilbake til Dashboard» på siste side i oppgavesettet. Du kan gjenåpne og redigere besvarelsen din så lenge prøven er åpen. Dersom ingen oppgaver er besvart ved prøveslutt, blir ikke besvarelsen din levert. Dette vil anses som «ikke møtt» til eksamen.

Trekk fra eksamen: Blir du syk under eksamen, eller av andre grunner ønsker å levere blankt/trekke deg, gå til «hamburgermenyen» i øvre høyre hjørne og velg «Lever blankt». Dette kan ikke angres selv om prøven fremdeles er åpen.

Tilgang til besvarelse: Du finner besvarelsen din i Arkiv etter at sluttida for eksamen er passert.

1 Oppgave 1 - Variabler og kontrollstrukturer (25%)

I denne oppgaven skal du programmere noe av funksjonaliteten til en reservasjons-applikasjon for kajakkutleiefirmaet *Kajakk og Kanoutleie*. Programmet består av et grafisk brukergrensesnitt som lar deg velge lånested, leveringssted, antall og type båter som ønskes leid, leietidspunkt og leveringstidspunkt. Basert på disse opplysningene blir så pris beregnet og reservasjonen presentert.

Kjeden Kajakk og Kanoutleie er forholdsvis lite utbredt og har kun 4 mulige steder hvor man kan leie og levere båter. Dette er Øya, Ila, Korsvika og Munkholmen. Dersom man leverer inn på et annet sted enn utleiestedet vil man få et gebyr på 50 kroner.

Kunden kan velge inntil 2 båter på en reservasjon og kan for hver av disse velge mellom tre ulike båttyper – Kano, Kajakk og Supboard. Hvor lenge kunden skal leie båten(e) bestemmer systemet fra lånetidspunkt og leveringstidspunkt.

Kunden skal også ha et eget tekstfelt som gir muligheten til å fylle inn ekstra kommentarer til utleieselskapet.

Pris for båtene per dag er:

- Kano: 395 kr
- Kajakk: 495 kr
- Supboard: 295 kr

Eksempel på utregning:

- Kano i 2 dager levert inn på samme sted vil koste: $395 \text{ kr} * 2 = 790 \text{ kr}$
- Supboard i 3 dager levert inn på annen plass vil koste: $295 \text{ kr} * 3 - 50 = 835 \text{ kr}$

Lag både HTML-koden for skjemaet og koden som må ligge bak knappen bestilling. Knappen beregner som sagt pris for oppholdet og presenterer den registrerte informasjonen. Dersom brukeren gir inn ugyldig informasjon i skjemaet skal systemet gi tilbakemelding om dette.

Skriv ditt svar her

1	
---	--

Maks poeng: 10

2 Oppgave 2 - Matriser (25%)

Tenk deg et sjakkbrett på 8x8 ruter kodet som en 2-dimensjonal matrise.

De forskjellige brikkene er representert ved forskjellige bokstaver. For eksempel er kongene kodet med bokstaven 'K'.

Vi bryr oss i denne oppgaven ikke om andre brikker, ei heller skiller vi mellom hvit eller sort konge. Figuren under viser en visualisering av denne matrisen med 2 konger plassert i hver sin rute.

	A	B	C	D	E	F	G	H
8								
7								
6					K			
5								
4								
3								
2		K						
1								

Skriv kode som går gjennom denne 2-dimensjonale matrisen for å finne rutene med konger. Det vil si ruter i den 2-dimensjonale matrisen som inneholder bokstaven 'K'.

Figuren over er kun et eksempel og vi vet ikke i hvilke ruter kongene befinner seg. Vi vet heller ikke hvor mange konger det er i matrisen. Det kan være både færre og flere enn de 2 i eksempelet.

Presenter så koordinatene til rutene hvor det befinner seg konger på følgende måte:

Konge E6

Konge B2

Skriv ditt svar her

1	
---	--

Maks poeng: 10

3 Oppgave 3 - Hendelser (25%)

Gitt følgende index.html fil:

```
<html>
  <head>
    <meta charset="UTF-8" />
  </head>
  <body></body>
  <script src="index.js"></script>
</html>
```

Skriv index.js slik at:

- Klokkeslettet vises på nettsiden.
 - Klokkeslettet skal oppdateres hvert sekund
- 2 sekunder etter nettsiden har blitt åpnet, skal en knapp vises som kan brukes til å endre bakgrunnsfargen til nettsiden:
 - Første gang knappen klikkes skal bakgrunnsfargen endres til rød
 - Andre gang knappen klikkes skal bakgrunnsfargen endres til hvit igjen
 - Tredje gang knappen klikkes skal bakgrunnsfargen endres til rød igjen
 - osv

Skriv ditt svar her

1	
---	--

Maks poeng: 10

4 Oppgave 4 - Funksjonelle algoritmer (25%)

Gitt tabellen:

```
let v = [2, 4, 6];
```

vil koden:

```
console.log("each element of v added by 2:", v.map(e => e + 2));
```

produsere utskriften:

```
each element of v added by 2: [4, 6, 8]
```

Bruk på tilsvarende måte funksjonelle algoritmer til å produsere følgende utskrifter:

a) each element of v multiplied by 3: [6, 12, 18]

b) each element index of v added by 2: [2, 3, 4]

c) elements of v larger than 3: [4, 6]

d) sum of element indexes of v: 3

Gitt tabellen:

```
let grades = [
  { student: 'Ola', grade: 'C' },
  { student: 'Kari', grade: 'B' },
  { student: 'Knut', grade: 'B' },
];
```

Bruk funksjonelle algoritmer til å produsere følgende utskrifter:

e) Did any student get D: no

f) Students that got B: ['Kari', 'Knut']

g) Ola's grade: C

Skriv ditt svar her

1	
---	--

Maks poeng: 10