

# EKSAMENSFORSIDE

## Skriftlig eksamen uten tilsyn

(Hjemmeeksamen)

Fylles ut av ansvarlig faglærer

Emnekode: PRG1100	Emnenavn: Grunnleggende programmering 2	
Campus: Ringerike	Fakultet: Handelshøyskolen	
Ansvarlig faglærer: Ståle Vikhagen		
Utlev. tidspkt: 25.5.2021 kl 9.00	Innlev. tidspkt: 25.5.2021 kl 13.00	Innlev. sted: Wiseflow
Skrifttype: --	Skriftstørrelse: --	Linjeavstand: --
Antall sider innledning / metatekst : --	Antall ord: --	Maks antall sider ekskl. forside og vedlegg: --
<p>Merknader:</p> <p>Selv om eksamen gjennomføres som hjemmeeksamen pga koronasituasjonen, så er det viktig å være klar over at det er <b>individuell hjemmeeksamen</b>. Alle former for samarbeid, deling osv er å betrakte som fusk og vil bli behandlet etter universitetets eksamensreglement. Alle trykte og elektroniske hjelpemidler er tillatt, men du må likevel følge reglene for kildehenvisning for å unngå fusk.</p> <p>Oppgavesettet består av <b>5 oppgaver og alle oppgavene skal besvares</b>. Innlevert dokument skal ha oppgavene med deloppgaver besvart i nummerrekkefølge. <b>Husk å kopiere kode fra kodeeditor til besvarelse/flow.</b></p> <p>Les gjennom hele oppgavesettet før du begynner å løse oppgavene. Les hver oppgave nøye slik at du svarer på det du blir spurt om. Vektingen av oppgaven viser hvor mye den teller ved sensuren.</p> <p><b>Legg spesielt merke til informasjonen om de sekvensielle tekstfilene oppgitt på side 3.</b></p>		

**Caset: Dekkhotell**

Firmaet DH er et nystartet dekkhotell som tilbyr pakkepriser på dekkskift, rengjøring og oppbevaring av dekk i personbilsegmentet.

En kunde kan ha avtale om dekk for flere biler ved dekkhotellet (dekksett) og en avtale for et dekksett kan bestå av flere oppbevaringer over tid. En kunde registreres ved første gangs omlegging av dekk, hvor en da registrerer kunden, dekksettet og oppbevaringen. Prisen som avtales gjelder fram til neste omlegging/dekkskift med oppbevaring.

## Filstruktur for sekvensielle tekstfiler, databasestruktur for databasen og krav til grafisk brukergrensesnitt

**For spørsmål hvor du skal programmere mot sekvensielle tekstfiler er filstrukturen følgende:**

Kunde, for hver kunde lagres, <Mobilnr, Fornavn, Etternavn, epost>

Dekksett, for hvert dekksett lagres, <Mobilnr, Regnr>

Oppbevaring, for hver oppbevaring lagres, <Mobilnr, Regnr, Innlevert, Utlevert, Hylle, Pris>

Utlevert i Oppbevaring får 'X' ved registrering av en ny oppbevaring, som endres til aktuell dato ved neste dekkskift.

### Krav og tips til filhåndteringen:

- I filene ligger det **ett felt pr linje**. Strukturen <Mobilnr, Fornavn, Etternavn, epost> er bare for å vise hva en post består av
- All filhåndtering skal være at en leser linje for linje («post for post») fra fil med test på eof og skriver linje for linje («post for post») til fil
- Filene er lagret i samme katalog som programmet
- Datoer registreres i formatet <år-måned-dag>, dvs 08022021 for 08.02.2021

**For spørsmål hvor du skal programmere mot database er databasestrukturen som følger:**

Kunde(Mobilnr, Fornavn, Etternavn, epost)

Dekksett(Mobilnr\*, Regnr)

Oppbevaring(Mobilnr\*, Regnr\*, Innlevert, Utlevert, Hylle, Pris)

Utlevert i Oppbevaring får NULL-merke ved registrering av en ny oppbevaring, som endres til aktuell dato ved neste dekkskift.

Det skal kobles mot databasen via **mysql.connector** etter følgende «koblingsstreng»:

- `mysql.connector.connect(host='localhost', port=3306, user=' Dekksjef ', passwd=' eksamen2021', db='Dekkkhotell')`

**Grafisk brukergrensesnitt, skal være basert på:**

- tkinter
- grid som geometry manager

## **Prosedural programmering, TUI-basert grensesnitt, mot filstruktur**

### **Spørsmål 1 (vekt 1):**

Lag et program som leser all informasjon om kundene inn i en to-dimensjonal liste, og som skriver ut mobilnr, fornavn og etternavn ved en gjennomgang av lista.

## **Prosedural programmering, GUI-basert grensesnitt, mot database**

### **Spørsmål 2 (vekt 2):**

Lag et vindu hvor du har en liste med alle oppbevaringer som ikke er avslutta (regnr og innlevert), sortert på innleveringstidspunkt, og når en velger i lista får en informasjon om hvem som er kunde for dekksettet og hvor det er plassert på dekkhotellet (mobilnr, etternavn, epost og hylle).

### **Spørsmål 3 (vekt 2):**

Lag et vindu for å registrere en ny kunde med dekksett og oppbevaring. Det må sjekkes at kunden ikke finnes fra før.

## Objektorientert programmering, TUI-basert grensesnitt, mot filstruktur

### Spørsmål 4 (vekt 1):

Gitt følgende klassediagram for klassen Kunde.

Kunde
<code>__mobilnr</code> <code>__fornavn</code> <code>__etternavn</code> <code>__epost</code>
<code>__init__()</code> <code>set_mobilnr(mobilnr)</code> <code>set_fornavn(fornavn)</code> <code>set_etternavn(etternavn)</code> <code>set_epost(epost)</code> <code>get_mobilnr()</code> <code>get_fornavn()</code> <code>get_etternavn()</code> <code>get_epost()</code>

Skriv koden for konstruktøren hvor metoden `__init__(mobilnr,fornavn,etternavn,epost)` instansierer et objekt med innlesing av dataattributter i et hovedprogram.

Skriv koden for metoden `set_epost`, basert på ny epost som inndata i variabel fra brukeren.

Skriv koden for metoden `get_epost`, hente ut/vise kundens epost.

### Spørsmål 5 (vekt 1):

Basert på klassediagrammet i oppgave 5, lag et program som leser og de-serialiserer objekter fra binærfile Kunde.dat og skriver ut mobilnr, etternavn og e-post for alle kundene. Dette skal gjøres for alle kundene i binærfile.