Bokmål

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i:	INF1000 — Grunnkurs i objektorientert programmering
Eksamensdag:	Fredag 4. desember 2015
Tid for eksamen:	14.30 (4 timer)
Oppgavesettet er på:	8 sider
Vedlegg:	Ingen
Tillatte hjelpemidler:	Alle trykte og skrevne

1. Kontroller at oppgavesettet er komplett og les nøye gjennom oppgavene før du løser dem.
2. Du kan legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens «ånd». Gjør i så fall rede for disse forutsetningene og antagelsene.
3. Poengangivelser øverst i hver oppgave angir maksimalt antall poeng. Sammenlagt gir alle oppgavene maksimalt **100** poeng. Unngå å bruke en stor del av tiden din på oppgaver som gir deg få poeng.
4. Svarene skal skrives på gjennomslagspapir. Skriv hardt nok til at besvarelsen blir mulig å lese på alle gjennomslagsarkene, og ikke legg andre deler av eksamensoppgaven under når du skriver.
5. Du beholder selv underste ark etter levering av de to øverste til eksamens-inspektøren.

Tilpasset IN1000 dvs noe Pythonifisert og noen temaer fjernet

Oppgave 1 (3 poeng)

a) Hva er verdien til **tall** etter at følgende kode er utført?

```
tall = 3 + 2 + 1
tall = tall * 2
```

b) Hva skrives ut på skjermen når følgende kode utføres?

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

```
a = 10
b = 1
while a > 0:
  b = b * 2
  a = a - b

print("a=" + a)
print("b=" + b)
```

Oppgave 2 (4 poeng)

Vi har en metode **doble** som vist nedenfor:

```
def doble(a):
    a = a * 2
    return a
}
```

a) Hva er verdien til **a** etter at følgende kode er kjørt?

```
a = 2
b = doble(a)
```

b) Hva er verdien til **b** etter at følgende kode er kjørt?

```
a = 2
b = doble(a + 1)
```

Oppgave 3 (4 poeng)

<Digital reprresentasjon: ikke pensum IN1000>

Oppgave 4 (7 poeng)

a) Skriv en funksjon **beregnAreal (lengde, bredde)** som regner ut arealet av et rektangel ved å multiplisere lengde og bredde, og returnerer resultatet.

b) Endre funksjonen slik at verdien -1 returneres hvis oppgitt lengde eller bredde er et negativt tall.

Oppgave 5 (5 poeng)

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

Hva skrives ut om du kjører dette programmet?

```
class MinKlasse :

  def __init__ (self, startTekst) :
    self._minVerdi = 2
    self._minTekst = startTekst


  def inkrementerVerdi(self) :
    self._minVerdi += 4


  def settTekst(self, tekst) :
    self._minTekst = tekst


  def hentBeskjed(self) :
    return self._minTekst + str(self._minVerdi)

mittObjekt = MinKlasse("AB")
tekst = mittObjekt.hentBeskjed()
mittObjekt.inkrementerVerdi()
mittObjekt.settTekst(tekst)
mittObjekt.inkrementerVerdi()
print(mittObjekt.hentBeskjed())
```

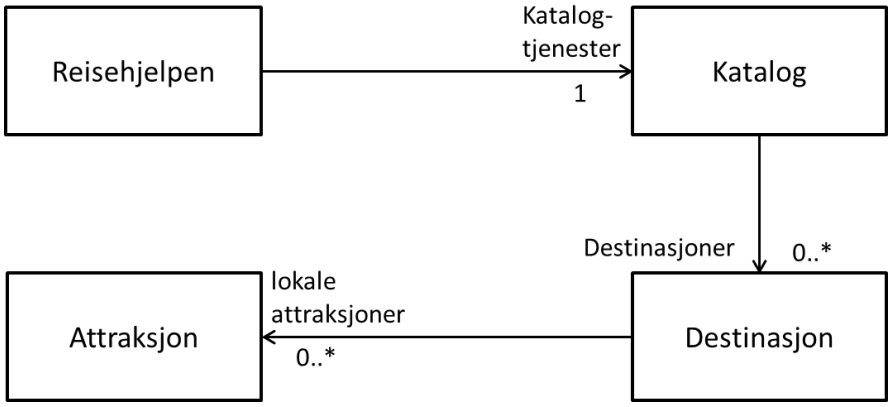
Oppgave 6 (5 poeng)

Skriv en funksjon **kampResultat(scoringerLagA, scoringerLagB)** som tar to heltall som parametre og returnerer en tekst**.** Funksjonen skal returnere "hjemme" dersom **scoringerLagA** er høyere enn **scoringerLagB**, "borte" dersom **scoringerLagB** er høyere enn **scoringerLagA**, og "uavgjort" dersom **scoringerLagA** er lik **scoringerLagB**.

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

Oppgave 7 (50 poeng)

Du er blitt bedt om å programmere deler av nettjenesten «Reisehjelpen» for planlegging av opplevelsesreiser for turister. Tjenesten bruker en klasse Katalog til å holde orden på destinasjoner og finne frem informasjon om de attraksjonene hver destinasjon kan by på. UML-diagrammet viser klassene som brukes av Reisehjelpen og deres relasjoner.



Om du hopper over en deloppgave er det likevel viktig at du leser hele teksten. Du kan fritt bruke klasser og metoder som er oppgitt tidligere i oppgaven selv om du ikke har skrevet dem.

Alle datoer representeres i programmet som heltall på formen aammdd der aa, mm og dd angir henholdvis årstall, måned og dag (4. desember 2015 lagres dermed som heltallet 151204).

Hver attraksjon har et navn, en sesongstart og en sesongslutt, og kan være egnet eller ikke for barn:

```
class Attraksjon :

  # Konstruktoer.
  # Parameteren barn tar en boolsk verdi: Egnet eller ikke for barn
  def __init__ (self, navn, barn, fra, til) :
    self._navn = navn

  # Skriver ut alle instansvariables innhold på en linje på terminal
  # (det er ikke vesentlig hvordan du velger å formattere dette)
  def skrivAttr (self) :

  # Returner en boolsk verdi - er attraksjonen egnet for barn?
  def forBarn (self) :

  # Returnerer om en attraksjon er aapen minst en av dagene i en gitt
  # periode (inkludert forste og siste dag i perioden):
  def aapenIPeriode (self, fra, til) :
```

a) 5 poeng
Skriv klassen **Attraksjon** med datarepresentasjon, konstruktør og metodene **skrivAttr** og **forBarn**.

b) 5 poeng
Skriv metoden **aapenIPeriode** i klassen **Attraksjon.**

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

Oppgave 7 (forts)

Klassen **Destinasjon** representerer destinasjoner denne tjenesten kjenner til. Hvert objekt av klassen **Destinasjon** har et navn og en samling attraksjoner som lagres i en **ArrayList**. Navnet og attraksjonene oppgis som parametere til konstruktøren når det opprettes en ny **Destinasjon**.

```
class Destinasjon :
  # Konstruktør skrives av deg

  # Returnerer destinasjonens navn:
  def hentDestNavn (self) :

  # Skriver ut sitt navn og alle sine attraksjoner på terminalen:
  def skrivDest (self) :

  # Legger til en ny lokal attraksjon:
  def leggTilAttr (self, nyAttr) :

  # Returnerer et tall som angir hvor mange av de lokale
  # attraksjonene som oppfyller et sett med krav (se deloppgave e):
  def antallAktuelleAttr (self, barn, fra, til) :
```

c) 3 poeng
Skriv konstruktør for klassen **Destinasjon.**

d) (7 poeng)
Skriv metodene **hentDestNavn**, **skrivDest** og **leggTilAttr** i klassen **Destinasjon.**

e) (5 poeng) (vanskelig)
Skriv metoden **antallAktuelleAttr (self, barn, fraDato, tilDato)** i klassen Destinasjon. Metoden skal returnere et tall som angir hvor mange av de lokale attraksjonene som oppfyller et sett med krav etter følgende regler:

- Hvis parameteren **barn** er **True** skal kun attraksjoner som egner seg for barn regnes med (alle egner seg for voksne)
- Kun attraksjoner som er åpne minst en dag i perioden fra og med **fraDato** til og med **tilDato** skal regnes med

- Eksamenssettet fortsetter på neste side - -

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

Oppgave 7 (forts)

Klassen **Katalog** holder orden på alle destinasjoner og tilbyr ulike operasjoner på disse:

```
class Katalog :

  # Konstruktør. Leser alle destinasjonene fra fil og oppretter,
  # objekter som legges inn i en Dictionary med destinasjonens navn
  # som nøkkel:
  def __init__ (self, katalogfil) :
    self._katalogfil = katalogfil
    self._destKatalog = {}
    self._lesDestinasjonsfil(self._katalogfil)

  # Metode som kalles fra konstruktøren for å lese inn katalogdata
  # i _destKatalog.
  # NB:  Denne skal du ikke skrive:
  def _lesDestinasjonsfil (self, filnavn) :

  # Metode som skriver ut navnet på alle destinasjoner i katalogen:
  def skrivDestListe (self) :

  # Metode som skriver ut navn og informasjon om alle attraksjoner
  # på den oppgitte destinasjonen:
  def skrivEnDest (self, destNavn) :

  # Metode som legger til en ny attraksjon for en destinasjon.
  # Om destinasjonen ikke finnes skal metoden returnere uten å
  # gjøre noe:
  def nyAttr (self, destNavn, attrNavn,bVennlig, apenFra, apenTil) :

  # Metode som leser nye attraksjoner for en destinasjon fra fil.
  # Om destinasjon med angitt navn ikke eksisterer fra før opprettes
  # en ny destinasjon, ellers legges attraksjonene til den
  # eksisterende destinasjonen. Filformat beskrives i deloppgave h):
  def nyDestFraFil (self, destNavn, filnavn) :

  # Metode som går gjennom alle destinasjoner, og returnerer navnet
  # på den destinasjonen som har flest attraksjoner som
  # tilfredsstiller krav som beskrevet i oppgave e):
  def finnBesteDest (self, barn, fra, til) :
```

- - **Eksamenssettet fortsetter på neste side** - -

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

Oppgave 7 (forts)

f) 5 poeng
Skriv metodene **skrivDestListe** og **skrivEnDest** i klassen **Katalog**.

g) 5 poeng
Skriv metoden **nyAttr** i klassen **Katalog**.

h) 7 poeng
Skriv metoden **nyDestFraFil** i klassen **Katalog.** Filen som skal leses inneholder en eller flere attraksjoner som tilhører samme destinasjon. For hver attraksjon ligger det *alltid* 4 linjer på følgende format, en linje for hver attributt. Eksempelet viser en fil for destinasjon Oslo:

```
Slottet
VOKSNE
150101
161231
Frognerbadet
BARN
150517
150820
```

i) 8 poeng (vanskelig)
Skriv metoden **finnBesteDest** i klassen **Katalog**.

- Eksamenssettet fortsetter på neste side - -

Tilpasset IN1000
dvs noe Pythonifisert og
noen temaer fjernet

Oppgave 8 (7 poeng)

<Personopplysningsloven: ikke pensum IN1000>

Oppgave 9 (15 poeng)

a) (7 poeng)
Skriv en funksjon **trimZeros (a)** som tar inn en liste med heltall og returnerer en liste med heltall hvor alle (eventuelle) nuller i starten og slutten av listen er fjernet. Dersom det er nuller inne i listen (dvs som har andre tall foran og bak seg) skal disse *ikke* fjernes. Gitt en liste [0,0,1,2,0,3,0,0,4,0] som argument, skal funksjonen altså returnere listen [1,2,0,3,0,0,4]. Effektiviteten av løsningen blir ikke tillagt vekt, formålet er kun at koden skal gi ønsket resultat.

b) (8 poeng)
Skriv en funksjon som tar inn en tekststreng som parameter og som returnerer et heltall. Funksjonen skal telle antall ulike bokstaver i den fikk inn som parameter og returnere dette antallet. Hvis funksjonen heter **telling**, så skal f.eks. setningen.

```
v = telling ("accag")
```

føre til at variabelen **v** blir tilordnet verdien 3.