



**Høgskolen i Telemark**

Fakultet for allmennvitenskapelige fag

**EKSAMEN**

**I**

**6108/6108N PROGRAMMERING I JAVA**

**21. 12. 2015**

Tid:	4 timer
Sidetall:	Framsida + 5 sider
Hjelpemiddel:	Alt trykt og skriftlig materiale.
Merknader:	Vekting er gitt ved starten av hver oppgave. Kandidaten kan som en del av besvarelsen bruke kall på metoder i Min-klassen (se vedlegg).
Vedlegg:	Min-klasse. Se side 5.

Sensuren publiseres via StudentWeb.

**Oppgave 1****\*\*\* FLERVALG \*\*\*****Vekting: 20%**

Angi spørsmålsnummer sammen med ditt svaralternativ, f.eks. slik: **1 a)**

Flervalgsdelen består av 10 spørsmål, der hvert spørsmål har kun ett riktig svaralternativ. Det er anledning til å gardere (dvs. angi flere svaralternativ). Riktig svaralternativ gir 3 poeng, (hvert) feil svaralternativ gir -1 poeng, mens et ubesvart spørsmål gir 0 poeng.

**NB!** Du kan forutsette at nødvendige import-setninger er på plass i alle kodeutsnittene.

1. Hva er verdien i variabelen `x` etter tilordningen dersom verdien i `y` er 13?

```
int x = 3 + y/4;
```

- a) 4
- b) 6
- c) 6.25
- d) Ingen verdi pga. syntaksfeil

2. Hva er verdien i variabelen `x` etter tilordningen dersom verdien i `y` er 13?

```
int x = 3 + y%4;
```

- a) 3
- b) 4
- c) 6.25
- d) Ingen verdi pga. syntaksfeil

3. Hva blir utskriften i konsollet dersom innholdet i `int`-variabelen `y` er lik 2?

```
out.println("x = " + (y + 5));
```

- a) `x = 7`
- b) `x = (7)`
- c) `x = 25`
- d) Ingen utskrift pga. syntaksfeil

4. Hva er verdien i variabelen `x` umiddelbart etter at **for**-løkka er avsluttet?

```
int x = 0;
for (int y=2; y<10; y++)
    if ( y%2 == 0 )
        x += y;
```

- a) 02468
- b) 10
- c) 20
- d) Ingen verdi pga. syntaksfeil

5. Hva blir skrevet i konsollet etter denne sekvensen av programsetninger?

```
double x = 0.0;
for (int i=1; i<y.length(); i++) // Tabellen y har int-verdier
    x += y[i]/2;                // i samtlige komponenter
out.println(x);
```

- a) Desimalt gjennomsnitt av alle tallene i y-tabellen
- b) Desimal verdi for halve summen av alle tallene i y-tabellen
- c) En feilmelding pga. semantisk feil
- d) En feilmelding pga. syntaksfeil

6. Hva blir skrevet i konsollet etter denne sekvensen av programsetninger?

```
int[] t = {3, 2, 1, 1};
out.println(t[1] / t[2] + t[t[0]]);
```

- a) 3
- b) 1
- c) En feilmelding pga. semantisk feil
- d) En feilmelding pga. syntaksfeil

7. Hva blir returverdien fra metodekallet `met(0.9)`? Metoden er gitt ved:

```
private void met(double t) {
    return -0.3 + t / 3;
}
```

- a) -0.3
- b) 0.0
- c) 0.2
- d) Ingen verdi pga. syntaksfeil

8. Hvor mange ganger blir setningen i denne løkke-kroppen utført?

```
int i = 0;
do { i = i*2; } while (i <= 10);
```

- a) 4
- b) 10
- c) Ukjent antall pga. evig løkke
- d) Ingen utførelser pga. syntaksfeil

9. Hvilket alternativ vil du si omtaler *konstanter* mest korrekt?

- a) Variabler som er låst mot verdiendring ved bruk av modifikatoren **final**
- b) Typisk benyttes konstanter til å nullstille tabeller
- c) Klasser med konstanter skal ikke ha `toString`-metoder
- d) Konstanter er synlige utenfor skopet til metoden der de er deklartert

10. Hvordan vil du fullføre setningen: «*En klasse ...*»

- a) kan ha flere **return**-setninger
- b) kan ikke ta mer enn én parameter
- c) kan ha flere konstruktører
- d) kan ikke ha **void**-metoder

\*\*\* SLUTT PÅ FLERVALG \*\*\*

**Oppgave 2****Vekting: 15%**

Skriv et komplett Java-program `Vekant` som leser inn et heltall i området 2 – 9. Dette heltallet (`antLinjer`) uttrykker antall linjer i en figur formet som bokstaven V, og denne skal skrives ut i konsollet ved bruk av de aktuelle linjenumrene 2 – 9.

**Eksempel** på tre ulike utskrifter: Én med antall linjer lik 2, én med 3, og så én med 6 linjer.

```

1 1      1 1      1      1
 2       2 2      2      2
          3       3      3
            4      4      4
              5 5      5
                6      6

```

Hver linje starter med innrykk (et antall blanke tegn), fortsetter med det sifferet som tilsvarer linjenummeret, fortsetter med blanke tegn som mellomrom, og avsluttes så med sifferet for linjenummeret. Innrykket øker (flere blanke tegn i starten), mens antall tegn i mellomrommet avtar, for hver linje nedover i figuren. Den siste linjen har kun én forekomst av linjenummeret, midtstilt i V-figuren.

**Oppgave 3****Vekting: 30%**

På sekvensielle tekstfiler er det lagret høydedata for elever på alle barneskolene i en kommune fra skoleåret 2014-15. Det er én fil for hver barneskole. Dataene er anonymisert slik at hver linje på en slik fil kun har lagret følgende data for en elev: *Kjønn* (kodet som **Jente/Gutt**), *klassetrinn* (1-7) og *høyde* i hele cm. På hver fil er dataene for en enkelt elev adskilt med bindestrek. Se et eksempel på figur A under til venstre.

Figur A

skole1.txt
J-4-132
G-3-129
G-1-118
J-7-150
G-2-123
: (Data linjene fortsetter)
:

Figur B

snittskole1.txt
1: J-118 G-119
2: J-123 G-121
:
:
6: J-144 G-143
7: J-150 G-148

Kommunens helseråd ønsker en oversikt over gjennomsnittshøydene (i hele cm) for hvert kjønn på hvert klassetrinn på en aktuell barneskole.

Skriv et komplett Java-program som leser inn filnavnet for en aktuell skole, går gjennom hele tekstfilen, og linje for linje henter ut de relevante dataene slik at de ulike gjennomsnittene kan beregnes. Til slutt skal programmet skrive ut de to snitthøydene (Jente-snitthøyde og Gutt-snitthøyde) for hvert enkelt klassetrinn, på hver sin linje, på en ny tekstfil med «snitt» satt foran navnet til inndatafila. Følg oppsettet på figur B ovenfor til høyre.

**Oppgave 4****Vekting: 10%**

I et Java-program inneholder tabellen `tab` objekt fra klassen `Spiller`. Tabellen er helt full, og har mer enn 300 `Spiller`-objekt. Et objekt fra klassen har følgende informasjon om en enkelt spill-deltager: Kallenavn (`String nick`), maksimum score (`int score`) i området 0-999 poeng, og antall gjennomførte spill (`int spill`).

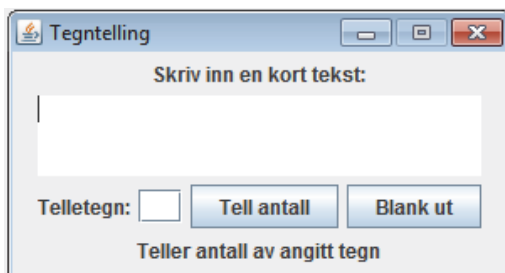
Det er ikke direkte tilgang til objektvariablene (innkapsling), så `Spiller`-klassen tilbyr `set`- og `get`-metoder på hver objektvariabel. Som vanlig tilbyr klassen en `toString`-metode for å vise tilstanden til spillerobjektet på formen: [`nick`, `score`, `spill`]. Klassen skal **ikke** programmeres!

Skriv **kun** de Java-setningene (ikke et komplett program) som er nødvendige for å skrive ut i konsollet en «topp-10-liste» over de beste spillerne etter score, med deres kallenavn, forutsatt at tabellen `tab` allerede er ferdig fylt.

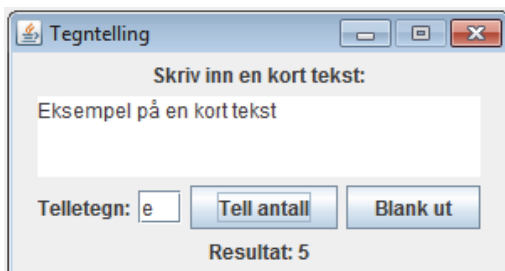
**Oppgave 5****Vekting: 25%**

Skriv et GUI-program som kan brukes til å hente ut informasjon fra en kort tekst skrevet inn av brukeren. Programmet skal kunne telle antall forekomster av ett enkelt tegn som brukeren angir.

Programmet skal ha et tekstområde (3 linjer med 20 kolonner), et tekstfelt (lengde 2) for å kunne skrive inn det tegnet som det ønskes opptelling av i teksten, en knapp for å gjennomføre tellingen og en knapp for å nullstille programmet (klargjøre det til å håndtere neste tekst).



300 x 160 piksler



I en etikett nederst i vinduet får brukeren tilbakemelding på den valgte opptellingen.

Se figurene til venstre som viser GUI-et før og etter telling av et tegn i en aktuell tekst.

NB! Tellingen skal ikke skille mellom store og små bokstaver.

— SLUTT —

**VEDLEGG**

```

/*
*****
* Min: Verktøyklasse - dvs. samling av klassemetoder. Det er lov å
* kalle på disse metodene fra program der det måtte være aktuelt.
*****
*/

import static javax.swing.JOptionPane.*;
import static java.lang.Integer.*;
import static java.lang.System.*;
import static java.lang.Math.*;

public class Min {

    // Metoden tegner en sekvens av tegn i konsollet
    public static void skrivTegn(char t, int antall) {
        for (int i=1; i<=antall; i++)
            out.print(t);
    }

    // Metoden avgjør om et tegn er en (engelsk) bokstav
    public static boolean erBokstav(char tegn) {
        return ('A' <= tegn && tegn <='Z') || ('a' <= tegn && tegn <='z');
    }

    // Metoden avrunder et tall til én desimal
    public static double avrund1(double tall) {
        return (int)(tall*10 + 0.5)/10.0;
    }

    // Metoden avrunder et tall til to desimaler
    public static double avrund2(double tall) {
        return (int)(tall*100 + 0.5)/100.0;
    }

    // Metoden leser inn et heltall i området (min-max)
    public static int lesHeltall(int min, int max) {
        int antall=0;
        do {
            String innTekst = showInputDialog("Gi heltall (" + min + "-" + max + "): ");
            antall = parseInt(innTekst);
            if ( antall < min || antall > max )
                showMessageDialog(null, "Ulovlig verdi!");
        } while (antall < min || antall > max);
        return antall;
    }

    // Metoden trekker et tilfeldig heltall i området min - max
    public static int trekkTall(int min, int max) {
        return min + (int)( random()*(max-min+1) );
    }
}

```