

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Eksamen i INF1000 — Grunnkurs i objektorientert programmering  
Eksamensdag: 10. juni 2005  
Tid for eksamen: 14.30–17.30  
Oppgavesettet er på 8 sider.  
Vedlegg: Ingen  
Tillatte hjelpemidler: Alle

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene.

### Innhold

<b>1 Kortsvarsoppgave</b> (vekt 30%)	side 1
<b>2 Geografikunnskaper</b> (vekt 30%)	side 4
<b>3 DNA-sekvenser</b> (vekt 40%)	side 5

### Oppgave 1 Kortsvarsoppgave (vekt 30%)

*Denne oppgaven består av flere små deloppgaver, som hver teller like mye i vurderingen.*

#### 1a

Studer koden nedenfor:

---

```
class SkrivHei {
    public static void main (String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 2; j++) {
                i = i + 4;
                System.out.println("hei");
            }
        }
    }
}
```

---

Hvor mange ganger blir teksten `hei` skrevet ut når programmet `SkrivHei` kjøres?

*(Fortsettes på side 2.)*

**1b**

Hva skrives ut på skjermen når programmet nedenfor kjøres?

---

```
class EnkelRegning {
    public static void main(String[] args) {
        int a = 4;
        int b = 9;
        a = b;
        b++;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

---

**1c**

Hva skrives ut på skjermen når programmet nedenfor kjøres?

---

```
class SkrivUt2{
    public static void main (String[] args){
        String s = "abcdefghijklm";
        String t = s.substring(0,3);
        System.out.println("t er nå:" + t);
        int lengde = s.length();
        int kvart = lengde/4;
        int halv = lengde/2;
        String u = s.substring(halv - kvart, halv + kvart);
        System.out.println("u er nå:" + u);
        if (s.indexOf("deF")>(-1)) {
            System.out.println("deF finnes");
        }
        else {
            System.out.println("deF finnes ikke");
        }
        if (s.endsWith("klm")){
            System.out.println("s ender med klm");
        }
    }
}
```

---

**1d**

Hva skrives ut på skjermen når programmet nedenfor kjøres?

---

```
class Beregninger{
    public static void main(String[] args){
        int x = 9;
        int y = 3;
```

(Fortsettes på side 3.)

```
int z = x - y;
System.out.println("verdien til z er " + z);
if ((x-z) == y){
    System.out.println("like");
}
else {
    System.out.println("ulike");
}
if ((x > 8) || (y < 2)){
    System.out.println("sann");
}
else{
    System.out.println("usann");
}
x++;
y += x;
int d = --x + y--;
System.out.println("d = " + d);
}
}
```

---

**1e**

Hva skrives ut på skjermen når programmet nedenfor kjøres?

```
class WhileTest {
    public static void main (String[] args) {
        boolean fortsett = true;
        int k = 3;
        while (fortsett) {
            k += 3;
            fortsett = !fortsett;
            System.out.println("k = " + k);
        }
    }
}
```

---

**1f**

Studer kodelinjene nedenfor:

```
int[] a = new int[50];
... // fyller inn tall i arrayen a
int s = 1;
for (int i = 0; i < a.length; ++i)
    s *= a[i];
}
```

---

Hva blir innholdet i variabelen s etter at denne koden er utført?

(Fortsettes på side 4.)

**1g**

Studér koden for klassen Person:

```
1 class Person [  
2     int navn;  
3     int alder;  
4     String adresse;  
5  
6     public Person (String n, int a, String adr) {  
7         navn = n;  
8         alder = a;  
9         adresse = adr;  
10    }  
11 ]
```

Finn og korriger alle feil i denne koden. Referer til linjenummer der du vil gjøre rettinger eller tillegg i koden.

**Oppgave 2 Geografikunnskaper (vekt 30%)**

*Denne oppgaven består av to deloppgaver, som hver teller like mye i vurderingen.*

En lærer i geografi vil teste sine elevers kunnskaper. Han har derfor laget et java-program hvor hver elev skal registrere navn på byer og land som hun kjenner til. For hvert sted (både by og land) skal det også registreres folketall.

Du skal hjelpe geografilæreren med å gjøre programmet hans ferdig. For å få oversikt over problemet, starter du med å sette deg inn i koden hans som består av de tre klassene vist i figurene 1, 2 og 3 på side 7–8.

**2a**

Lag et UML klassediagram av programmet som består av de tre klassene vist i figurene 1, 2 og 3 på side 7–8.

**2b**

Programmet mangler utskriftrutiner slik at læreren fort kan få oversikt over hvilke opplysninger en elev har registrert.

Lag en utskriftrutine som skriver ut opplysninger om alle byer i hvert land under hverandre på skjermen, slik det er illustrert i eksempelet nedenfor. For land der det ikke er registrert tilhørende byer, skrives bare landsinformasjonen ut. For å løse dette problemet må du skrive kode i klassene By, Land og Geografikunnskaper.

*(Fortsettes på side 5.)*

Eksempel på utskrift:

---

Norge	4525000
*****	
Oslo	530000
Moss	26000
Narvik	19000
Hamar	26500
*****	
Sverige	9040000
*****	

---

...

---

### Oppgave 3 DNA-sekvenser (vekt 40%)

*NB! Legg merke til at i denne oppgaven teller deloppgavene a og b 10% hver, mens punkt c teller 20%.*

Denne oppgaven tar utgangspunkt i et problem hentet fra molekylærbiologi, men for å løse oppgaven trenger du kun å tenke på tekststrenger bestående av et gitt antall symboler (tegn).

#### 3a (vekt 10%)

DNA-sekvenser representeres ved hjelp av 4 symboler: A, T, C og G. I mange tilfeller vil det være interessant å vite den relative frekvensen til hvert av symbolene i en DNA-sekvens. Du skal nå lage en metode som teller opp antall forekomster av hvert symbol i en gitt DNA-sekvens og regner ut den tilhørende relative frekvensen.

Eksempel: Gitt DNA-sekvensen AATGGATC. Denne sekvensen består av 8 symboler, hvorav 3 forekomster av A. Dermed er den relative frekvensen av A i sekvensen  $3/8 = 0,375$ .

Skriv metoden `symbolFrekvens` som tar en `String` som parameter og returnerer en array med 4 verdier (frekvensen til A, frekvensen til T, frekvensen til C, frekvensen til G):

---

```
double[] symbolFrekvens(String sekvens) {
    ...
}
```

---

#### 3b (vekt 10%)

Lag en klasse `DNAsekvens` som inneholder følgende informasjon:

- navnet på sekvensen,

(Fortsettes på side 6.)

- selve sekvensen,
- lengden på sekvensen (antall symboler),
- frekvensen av A, T, C og G.

Lag en konstruktør i klassen slik at et nytt DNAsekvens objekt kan opprettes ved kodesetningen

```
DNAsekvens a = new DNAsekvens(navn,sekvens);
```

der argumentene navn og sekvens begge er av type String.

### 3c (vekt 20%)

Anta at du har en fil "DNA.txt" hvor første linje forteller hvor mange DNA-sekvenser det er i filen, mens hver av de resterende linjene i filen beskriver en sekvens. Hver linje starter med navnet på sekvensen, deretter et mellomrom (ett eller flere blanke tegn), og så selve sekvensen.

Eksempel på filen "DNA.txt":

---

```
137
AY1231  AAATCAGAAG
AY5432  GGAATCCAGTAAAA
AY3234  GGAGTCGATGA
AY3423  GGAGTCGATCCCAGA
...
```

---

Lag en metode

```
DNAsekvens[] lesSekvenserFraFil (String filnavn) {
    ...
}
```

som leser inn alle DNA-sekvensene i en array bestående av DNAsekvens-objekter. Denne arrayen returneres fra metoden.

(Fortsettes på side 7.)

## Programkode for oppgave 2

---

```
1 import java.util.*;
2
3 class Geografikunnskaper {
4     HashMap LandOversikt = new HashMap();
5
6     void leggTilLand(String navn, int innbyggertall) {
7         Land land = new Land(innbyggertall);
8         LandOversikt.put(navn, land);
9     }
10
11    void leggTilByILand(String navnLand, String navnBy,
12                       int innbyggertall, boolean hovedstad) {
13        boolean finnes = LandOversikt.containsKey(navnLand);
14        if (!finnes) {
15            Land land = new Land(0);
16            LandOversikt.put(navnLand, land);
17        }
18
19        Land nland = (Land) LandOversikt.get(navnLand);
20
21        nland.leggTilBy(navnBy, innbyggertall, hovedstad);
22    }
23
24    public static void main(String args[]) {
25        Geografikunnskaper geo = new Geografikunnskaper();
26
27        geo.leggTilLand("Norge", 4525000);
28        geo.leggTilLand("Sverige", 9040000 );
29        geo.leggTilByILand("Norge", "Oslo", 530000, true);
30        geo.leggTilByILand("Norge", "Moss", 26000, false);
31        geo.leggTilByILand("Norge", "Narvik", 19000, false);
32        geo.leggTilByILand("Norge", "Hamar", 26500, false);
33
34        // Anta at programmet senere skal utvides slik at elever
35        // kan legge inn informasjon om nye land og byer ved bruk
36        // av meny og inntasting på skjermen. NB! Dette skal ikke
37        // lages nå.
38
39        // Her skal du sørge for at alle byer i alle land blir
40        // skrevet ut på skjermen.
41    }
42 }
```

---

Figur 1: Klassen Geografi kunnskaper i Oppgave 2.

(Fortsettes på side 8.)

---

```
1 class Land{
2     int innbyggertall;
3     HashMap Byer = new HashMap();
4
5     Land(int innbyggertall) {
6         this.innbyggertall = innbyggertall;
7     }
8
9     void leggTilBy(String navn, int innbyggertall,
10                  boolean hovedstad) {
11         By by = new By(innbyggertall, hovedstad);
12         Byer.put(navn, by);
13     }
14
15     void skrivUtAlleLand() {
16         // Denne metoden skal du skrive i oppgave 2b
17     }
18 }
```

---

Figur 2: Klassen Land i Oppgave 2.

---

```
1 class By{
2     int innbyggertall;
3     boolean hovedstad;
4
5     By(int innbyggertall, boolean hovedstad) {
6         this.innbyggertall = innbyggertall;
7         this.hovedstad = hovedstad;
8     }
9
10    void skrivUt() {
11        // Denne metoden skal du skrive i oppgave 2b
12    }
13 }
```

---

Figur 3: Klassen By i Oppgave 2.

**SLUTT PÅ OPPGAVESETTET**