

**OPPGAVEN BLIR GJENNOMGÅTT PÅ FORELESNINGEN 21/10. DU BØR FORBEREDE DEG TIL FORELESNINGEN VED Å SETTE AV MINST TRE-FIRE TIMER TIL Å PRØVE Å LØSE OPPGAVEN PÅ EGENHÅND.**

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Tenkt eksamen i :           INF1000 — Grunnkurs i objektorientert programmering  
Gjennomgås:                På forelesningen tirsdag 21. oktober 2008 kl 14-16  
Oppgavesettet er på :       12 sider  
Vedlegg :                    Ingen  
Tillatte hjelpemidler :     Alle trykte og skrevne

- Dersom du savner opplysninger i noen av oppgavene, kan du selv legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i såfall rede for forutsetningene og antagelsene du gjør.
- Svarene kan skrives på disse oppgavearkene. I de oppgavene hvor det skal skrives programkode, anbefales det at du først skriver en kladd på eget ark før du fører svaret inn i disse oppgavearkene på avsatt plass.
- Noen av spørsmålene er flervalgsoppgaver. På slike oppgaver gis det til eksamen poeng etter hvor mange korrekte svar du gir. Du ville til eksamen ikke fått poeng hvis du hadde latt være å besvare et spørsmål, eller dersom du hadde krysset av begge svaralternativer.
- Hvis du til eksamen setter et kryss i en avkrysningsboks og etterpå finner ut at du ikke ønsket å krysse av der, kan du skrive "FEIL" like til venstre for den aktuelle avkrysningsboksen.
- Ikke bruk datamaskin når du løser oppgavene – det er bare å lure seg selv siden poenget er at denne prøven skal være så lik en ordentlig eksamen som mulig.

### Oppgave 1

Er disse programsetningene lovlige i Java?

JA	NEI
<input type="checkbox"/>	<input type="checkbox"/> <code>int i, j = i+1;</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>int j = 3, while (j--&gt;0) {i = j;}</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>int[]int[] int2 = new int[8]int[9];</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>boolean[] b = null;</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>boolean[] b = new boolean[true];</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>boolean[] b = false;</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>int 12X = (int) (true == false);</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>double x = (int) 3.5;</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>int j = 5, k = (int)(j/2.5);</code>
<input type="checkbox"/>	<input type="checkbox"/> <code>boolean c = (true != true) &amp;&amp; false;</code>

## Oppgave 2

a) Hvor mange ganger blir "INF1000" skrevet ut av følgende løkke:

```
int i = 1;
while (++i < 10) {
    System.out.println("INF1000");
}
```

Svar: .....

b) Hvor mange ganger blir "INF1000" skrevet ut av følgende løkke:

```
int i=1, j=1;
do {
    System.out.println("INF1000");
    i = i + 1;
    j = j + i;
} while (j * i < 100);
```

Svar: .....

c) Hvilken verdi får variabelen **resultat** her?

```
int resultat = 0;
for (int i=0; i<100; i=i+10) {
    for (int j=i; j<i+10; j++) {
        resultat++;
    }
}
```

Svar: .....

d) Hvilken verdi får variabelen **resultat** her?

```
int resultat = 0;
for (int i=0; i<10; i++) {
    for (int j=0; j<10; j++) {
        if (i == j-1) {
            resultat++;
        }
    }
}
```

Svar: .....

e) Hva blir skrevet ut på skjermen her?

```
String s1 = "10" + "20" + "30";
String s2 = "10 + 20 + 30";
String s3 = "10" + 20 + 30;
if (s1.equals(s2)) {System.out.println("A");}
if (s1.equals(s3)) {System.out.println("B");}
if (s2.equals(s3)) {System.out.println("C");}
```

Svar: .....

### Oppgave 3

Anta at følgende er deklarerert i en metode i et program:

```
int[][] tabell = new int[100][100];
```

Anta at arrayen **tabell** er fylt opp med verdier (positive og negative) som er lest fra en fil. Hvilke av alternativene under vil alltid gi som resultat at **max** inneholder maksimumsverdien i **tabell**?

- | JA                       | NEI                      |   |
|--------------------------|--------------------------|---|
| <input type="checkbox"/> | <input type="checkbox"/> | <pre>int k = 0; int max = 0; while (k &lt; tabell.length) {     if (max &lt; tabell[k]) max = tabell[k];     k++; }</pre>   |
| <input type="checkbox"/> | <input type="checkbox"/> | <pre>int max = tabell[0][0]; for (int i=0; i&lt;100; i++) {     for (int j=0; j&lt;100; j++) {         if (max &lt; tabell[i][j]) max = tabell[i][j];     } }</pre>   |
| <input type="checkbox"/> | <input type="checkbox"/> | <pre>int max = tabell[0][0]; for (int i=0; i&lt;100; i++) {     for (int j=0; j&lt;100; j++) {         if (max &lt; tabell[i][j] &amp;&amp; max &lt; tabell[j][i]) {             max = Math.max(tabell[i][j], tabell[j][i]);         }     } }</pre>  |
| <input type="checkbox"/> | <input type="checkbox"/> | <pre>int max = tabell[0][0]; for (int i=0; i&lt;100; i++) {     for (int j=i; j&lt;100; j++) {         int m = Math.max(tabell[i][j], tabell[j][i]);         max = Math.max(max, m);     } }</pre>  |
| <input type="checkbox"/> | <input type="checkbox"/> | <pre>int[] maxv = new int[100]; for (int i=0; i&lt;100; i++) {     maxv[i] = tabell[i][0];     for (int j=1; j&lt;100; j++) {         if (maxv[i] &lt; tabell[i][j]) {             maxv[i] = tabell[i][j];         }     } } int max = maxv[0]; for (int i=0; i&lt;100; i++) {     if (max &lt; maxv[i]) max = maxv[i]; }</pre> |



## Oppgave 6

Vi kan lage en sekvens med heltall på følgende måte: la de to første elementene i sekvensen begge være 1 og la deretter hvert element være summen av de to foregående. Den sekvensen vi da får starter slik: 1, 1, 2, 3, 5, 8, 13, ..... og kalles Fibonacci-sekvensen. Du skal nå lage en metode som beregner de  $n$  første elementene i Fibonacci-sekvensen og returnerer disse i en heltallsarray av lengde  $n+1$  (det første elementet i arrayen, svarende til posisjon 0, skal ikke benyttes). Posisjon  $k$  i arrayen som returneres skal inneholde det  $k$ 'te elementet i Fibonacci-sekvensen for  $k=1, \dots, n$ . Du kan anta at  $n \geq 2$ .

Svar:

```
int[] fibonacci (int n) {
```

```
}
```



## Oppgave 9

Anta at følgende program utføres:

```
class StudentSystem {
    public static void main (String[] args) {
        Studieprogram inf = new Studieprogram("Informatikk");
        Studieprogram bio = new Studieprogram("Biologi");

        new Student("Kari Olsen", inf);
        new Student("Eva Larsen", inf);
        new Student("Per Jensen", bio);

        bio.liste();
        inf.liste();
    }
}

class Studieprogram {
    String tittel;
    Student[] medlemmer;
    int antall;

    Studieprogram (String tittel) {
        this.tittel = tittel;
        medlemmer = new Student[100];
        antall = 0;
    }

    void meldInnStudent(Student stud) {
        medlemmer[antall] = stud;
        antall++;
    }

    void liste() {
        for (int i=0; i<antall; i++) {
            System.out.println(memlemmer[i].somTekst());
        }
    }
}

class Student {
    String navn;

    Student (String navn, Studieprogram sp) {
        this.navn = navn;
        sp.meldInnStudent(this);
    }

    String somTekst() {
        return navn;
    }
}
```

Skriver programmet noe ut på skjermen, og isåfall hva?

Svar:

## Oppgave 10

Anta at du har deklarerert en HashMap:

```
HashMap<String,String> cdSamling = new HashMap<String,String>();
```

og at du legger inn informasjon om CD'ene dine i HashMapen, med platens tittel som nøkkel og artistnavnet som verdi (du antar at alle platene dine alle har ulik tittel). Et eksempel er

```
cdSamling.put("Not going under", "Maria Arredondo");
```

Skriv noen programsetninger som:

- først ber om og leser inn et artistnavn fra tastatur (f.eks. "Maria Arredondo").
- deretter går gjennom HashMapen og skriver ut titlene til alle platene du har registrert med denne artisten. Hint: husk at du vil ha tak i både nøkklene og verdiene når du løper gjennom HashMapen. Det kan derfor lønne seg å lage en Iterator over nøklene (platetitlene).

Du kan anta at programsetningene plasseres slik i programmet at de har tilgang til HashMapen `cdSamling`.

Svar:



## Oppgave 11

I programmet nedenfor skal du lage en konstruktør til klassen Student. Konstruktøren skal ha studentens navn som parameter og skal initiere objektvariabelen `navn`. Du skal også skrive en objektmetode `void økPoeng(int poeng)` i klassen Student som øker antall studiepoeng med parameterens verdi. I klassen StudentTest skal du så sørge for å få initiere arrayen `uioStud` med 32000 Student-objekter. Studentobjektene i arrayen `uioStud` skal ha hvert sitt studentnavn `Stud-1`, `Stud-2`, `Stud-3`, .... `Stud-32000`. Dermed skal f.eks. `uioStud[252]` peke på et Student-objekt hvor studentens navn er satt lik `Stud-253`. Til sist skal du sette antall studiepoeng til 30 for hver av studentene `Stud-1`, ...., `Stud-25000`.

```
class Student {
    String navn;
    int antallStudiepoeng = 0;

    // Her skal du skrive konstruktøren

    // Her skal du skrive objektmetoden økPoeng

}

class StudentTest {
    Student[] uioStud = new Student[50000];

    public static void main (String[] args) {
        // Her skal du opprette 32000 Student-objekter med navn Stud-1,
        // Stud-2,...,Stud-32000 og legge dem inn i uioStud-arrayen

        // Her skal du øke antall studiepoeng med 30 for studentene
        // med studentnavn Stud-1, Stud-2, ...., Stud-25000.

    } // end main
}
```

## Oppgave 12

Filen **SpamOrd.txt** inneholder endel *spam-ord* (atskilt av blanke tegn), dvs ord som du ikke forventer å finne i seriøse eposter til deg, men som ofte forekommer i useriøse eposter (såkalt spam-epost). Du skal lage et fullstendig program som leser **SpamOrd.txt** og som deretter leser file **Epost.txt** og sjekker om denne inneholder noen av spam-ordene. Programmet skriver til slutt ut på skjermen om file **Epost.txt** inneholdt noen spam-ord eller ikke (du trenger ikke å skrive ut hvilke eller hvor mange spam-ord file inneholdt). Du kan anta at **SpamOrd.txt** maksimalt inneholder 200 ord. I denne oppgaven kan du godt legge all programkoden inn i main-metoden.

Svar:

(forts. neste side)



### Oppgave 13

Skriv metoden under som finner og skriver ut på skjerm hvor mange ulike verdier du har i en array og returnerer dette antallet. Hint: definer en heltallsvariabel **antUlike** og initier denne til 0. Løp deretter gjennom alle elementene i arrayen. For hvert verdi du ser øker du **antUlike** med 1 dersom det er siste gang denne verdien forekommer i arrayen, dvs hvis ingen elementer med høyere indeks er lik denne verdien (du kan sjekke dette med en ny løkke inni den ytre løkken). Dermed er du sikret at hver verdi bare telles en enkelt gang. Du kan anta at lengden av arrayen **a** er minst 1.

Svar:

```
void finnAntallUlikeTall (int[] a) {
```

```
}
```