UNIVERSITETET I OSLO

Kandidatnr

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF1000 — Grunnkurs i objektorientert programmering
Eksamensdag:	Onsdag 1. desember 2010
Tid for eksamen:	14.00 – 18.00
Oppgavesettet er på 16 sider.	
Vedlegg:	Ingen
Tillatte hjelpemidler:	Alle trykte og skrevne

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

- Les nøye gjennom hver oppgave før du løser den. For hver oppgave er det angitt det maksimale antall poeng du kan få hvis du svarer helt riktig. Summen av poengene er 100, slik at f.eks 5 poeng tilsvarer 12 minutter, 10 poeng tilsvarer 24 minutter, osv. (hvis du regner med å komme igjennom alt). Pass på at du bruker tiden din riktig. Setter du av 20 minutter til 10 poeng, får du 40 minutter til å se over alt til slutt!
- Kontroller også at oppgavesettet er komplett før du begynner å besvare det. Dersom du savner opplysninger i oppgaven, kan du selv legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i så fall rede for forutsetningene og antagelsene du gjør.
- Dine svar skal skrives på disse oppgavearkene, og ikke på separate ark. Dette gjelder både spørsmål med avkrysningssvar og spørsmål hvor du bes om å skrive programkode. Det er satt av plass i oppgavesettet slik at du kan skrive inn svarene dine.
- Får du likevel ikke plass nok i feltene i oppgaveteksten, kan du fortsette på eget ark.
- Noen av spørsmålene er flervalgsoppgaver. På disse oppgavene får du poeng etter hvor mange korrekte svar du gir. Du får *ikke* poeng hvis du lar være å besvare et spørsmål, eller dersom du krysser av begge svaralternativer.
- Hvis du har satt 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.
- Husk å skrive såpass hardt at besvarelsen blir mulig å lese på alle gjennomslagsarkene, men *ikke legg andre deler av eksamensoppgaven under når du skriver.*

(Fortsettes på side 2.)

Innhold

Lykke til!

(Fortsettes på side 3.)

Oppgave 1 Løkker (3 poeng)

Hvor mange ganger blir "Juleferie" skrevet ut?

1a

```
for (int i=10; i<=12; i++) {
  for (int j=0; j<6; j++) {
    System.out.println("Juleferie");
  }
}
```

Svar: _____

1b

```
for (int k=4; k>0; k--) {
  for (int j=0; j<k; j++) {
    System.out.println("Juleferie");
  }
}
```

Svar: _____

1c

```
for (int i=2; i<5; i*=2) {
  for (int k=i; k<5; k*=-2) {
    System.out.println("Juleferie");
  }
}
```

Svar: _____

(Fortsettes på side 4.)

Oppgave 2 Array og løkker (6 poeng)

Anta følgende deklarasjoner:

```
final int N = 5;
final int M = 5;
int[][] tall = new int[M][N];
int sum;
```

Anta videre at alle plassene i arrayen `tall` har fått tilordnet en **int**-verdi. For hver av kodebitene nedenfor, vil variabelen `sum` alltid inneholde summen av tallverdiene i `tall` etter endt kjøring?

Ja	**Nei**	

☐	☐	`sum = 0;`
		`int i = N;`
		`int j = M;`
		`while( i>0 && j>0 )`
		`  sum += tall[--i][--j];`

☐	☐	`sum = 0;`
		`for(int i=0; i<N; i++)`
		`  for(int j=0; j<tall[i].length; j++)`
		`  for(int k=tall[i][j]; k>0; sum++){`
		`  k--;`
		`  }`

☐	☐	`HashMap<String,String> h = new HashMap<String,String>();`
		`for(int i=0; i<N; i++)`
		`  for(int j=0; j<M; j++)`
		`  for(int k=0; k<tall[i][j]; k++)`
		`  h.put("tall"+i+j+k,null);`
		`sum =  h.size();`

☐	☐	`sum = 0;`
		`int i = N-1;`
		`while(i>=0){`
		`  int j = M;`
		`  while(j>0)`
		`  sum +=  tall[i][--j];`
		`  i--;`
		`}`

(Fortsettes på side 5.)

Oppgave 3 Java-syntaks (4 poeng)

Er disse uttrykkene lovlige programsetninger i Java?

Ja	Nei	
☐	☐	`enum Himmelretning = { NORD, ØST, SØR, VEST };`
☐	☐	`int[ ] binaryNumber = 0100110001;`
☐	☐	`boolean b = (true == false);`
☐	☐	`boolean b = b;`
☐	☐	`int tabell = new int[3][2];`
☐	☐	`String string = "String string";`
☐	☐	`double[ ] d = new double[3.2];`
☐	☐	`Hashmap<int,String> h = new Hashmap<int,String>();`

Oppgave 4 String (12 poeng)

I denne oppgaven kan det være nyttig å bruke String-metodene **public char** `charAt(int index)` og **public int** `length()`.

4a Sjekke forekomster av tegn i tekst (4 poeng)

Metoden under, som du skal kode, skal returnere **true** hvis tegnet `t` forekommer i `String`-variabelen `tekst`, og **false** ellers.

```
boolean forekommer(char t, String tekst){




}
```

(Fortsettes på side 6.)

4b Fjerning av repeterte tegn (4 poeng)

Skriv kode til metoden under. Den skal returnere en streng der alle repetisjoner av tegn i String-variabelen `tekst` er fjernet. For eksempel skal `utenRepetisjon("aababbabbac")` returnere `"abc"`. *Hint:* Bruk metoden i oppgave 4a.

```
String utenRepetisjon(String tekst){









}
```

4c Antall forskjellige tegn (4 poeng)

Skriv kode til metoden under. Den skal returnere antall forskjellige tegn som forekommer i String-variabelen `tekst`. For eksempel skal metoden returnere 3 hvis den kalles med String-verdien `"aababbabbac"`. *Hint:* Bruk metoden i oppgave 4b.

```
int antallForskjellige(String tekst){









}
```

(Fortsettes på side 7.)

Oppgave 5 Metoder og objekter (15 poeng)

5a Dato-klasse (7 poeng)

Du skal i denne oppgaven implementere to metoder i en klasse for dato som står beskrevet i kommentarene over deres signatur (signatur = første linjen av metoden).

```
class Dato{
  int dag;
  int mnd;
  int aar;

// Konstruktør
Dato(int dag, int mnd, int aar){









}

// Returnerer negativt tall hvis this kommer før d, 0 hvis this
// og d er samme dag, og positivt tall hvis this kommer etter d
// (this refererer til objektet som sammenlignes med d)
int sammenlign(Dato d){

















}
```

(Fortsettes på side 8.)

5b Testprogram (8 poeng)

Lag et fullstendig testprogram (inkludert main-metode og evt. import-setninger) som tester Dato-klassen. Du kan anta at testprogrammet ditt ligger på samme katalog som `Dato`-klassen. Du skal opprette to forskjellige objekter for 1. desember 2009 og et objekt for 1. november 2010. Deretter skal du kalle `sammenlign`-metoden for tre tilfeller der den skal returnere henholdsvis et negativt tall, 0 og et positivt tall. Disse verdiene skal du skrive ut til skjerm.

Oppgave 6 Et lite kundearkiv (20 poeng)

I denne oppgaven skal du programmere et kundearkiv for en lite selskap i telekom-bransjen. Arkivet er bygget opp om tre klasser som er nærmere beskrevet under: `Tlfnr`, `Kunde` og `Kundearkiv`.

6a Tlfnr og Kunde (5 poeng)

```
class Tlfnr{
  String nr;
  Dato opprettet;

  \\ Her kommer konstruktør og evt. andre metoder
}
```

(Fortsettes på side 9.)

`Tlfnr`-klassen har datastruktur som over. Telefonnummeret er her gitt som en String. Dato-klassen er som beskrevet i forrige oppgave og du kan anta at den inneholder metoder som beskrevet der selv om du ikke skulle ha implementert disse. Variabelen `opprettet` registrerer når kunden fikk nummeret. Vi antar for enkelhets skyld at kunden aldri sier opp et nummer.

Hver kunde er registrert med et entydig kundenummer og navn og adresse. Hver kunde har minst ett telefonnummer, kanskje flere. Disse er lagt i en fylt array (dvs. en array der ingen av feltene er **null**):

```
class Kunde{
  int knr;
  String navn;
  String adr;
  Tlfnr[ ] tlf;

  \\ Konstruktør og andre metoder
}
```

Du skal nå kode en metode som skal ligge i `Kunde`-klassen og ta inn et `Dato`-objekt som parameter. Metoden skal skrive til skjerm alle telefonnummer som var registrert på kunden på den angitte dato (dvs som var opprettet før den angitte dato).

(Fortsettes på side 10.)

6b Kundearkiv (15 poeng)

Klassen `Kundearkiv` skal kunne lagre inntil 10 000 Kundeobjekter. Disse skal lagres i en array som holdes sortert på kundenummer. De fleste av metodene i klassen skal vi se bort fra i denne oppgaven, men du skal gjøre følgende:

- Deklarere datastrukturen (arrayen og evt. andre variable)
- Kode en metode som tar et `Kunde`-objekt som parameter og setter det inn i arrayen. Husk at arrayen er sortert og skal være sortert også etter at du har satt inn `Kunde`-objektet.
- Kode en metode som tar et kundenummer som parameter og returnerer korresponderende Kunde-objekt hvis det finnes, og **null** ellers. Metoden har signatur `Kunde finnKunde(int knr);`
- Kode en metode som skriver til skjerm alle telefonnummer registrert i systemet på en angitt dato.

```
class Kundearkiv{




}
```

(Fortsettes på side 11.)

(Fortsettes på side 12.)

Oppgave 7 UML (10 poeng)

Lag både et UML klassediagram (uten variable og metodenavn) og et objektdiagram for Kundearkiv-systemet du har implementert i forrige oppgave.

(Fortsettes på side 13.)

Oppgave 8 Oppslag og vedlikehold (25 poeng)

8a Raskere oppslag ved Hashmap (10 poeng)

Datastrukturen i `Kundearkiv`-klassen er ikke egnet for raskt oppslag ut fra telefonnummer. For å få til det skal du legge til en Hashmap til `Kundearkiv`-klassen som forbinder et telefonnummer med en kunde. Du skal:

- Deklarere Hashmapen
- Skive en metode som legger inn forbindelser mellom alle telefonnummer og deres tilsvarende `Kunde`-objekter
- Implementere en metode med signatur `Kunde finnKunde(Tlfnr t);` som bruker Hashmapen til å finne frem til riktig kunde

(Fortsettes på side 14.)

8b Slå sammen kundeobjekter (15 poeng)

Inntil nå kan samme person stå oppført med flere kundenummer i `Kundearkiv`-klassen fra Oppgave 6b. Dette skal du rydde opp i på følgende måte:

- Alle `Kunde`-objekter med samme navn og adresse skal slås sammen til ett.
- For å bestemme kundenummeret til det nye objektet skal du velge det av kundens gamle kundenumre som har vært i bruk lengst. Dette finner du ved å se på når kundens telefonnumre ble opprettet.
- Alle kundens telefonnumre skal samles i det nye objektet.
- Alle forbindelsene i Hashmapen skal oppdateres.

Du kan anta at arrayen med kunder ligger sortert på kundenummer. Der det er flere objekter med samme navn og adresse, kan det være lurt å opprette et helt nytt `Kunde`-objekt og vente med å slette de gamle til all informasjonen er på plass i det nye!

(Fortsettes på side 15.)

(Fortsettes på side 16.)

Oppgave 9 Personvern (5 poeng)

Et telefonfirma ønsker å selge følgende informasjon om kundene sine til andre firma: navn, adresse, personnummer og om det er noen betalingsproblemer med denne kunden. Er dette lovlig? Begrunn svaret med henvisning til aktuelle paragrafer i Lov om behandling av personopplysninger (personopplysningsloven)som som er gjennomgått i pensum og hvorfor disse paragrafene evt. omtaler denne bruken av persondata.