



## Oppgave 1 – 15% (~ 36 minutter. Deloppgaver teller 5%)

a) Hva blir skrevet ut på skjermen når denne koden blir utført:

```
public class HvaSkjer1 {  
  
    public static void main(String[] args) {  
        int a = 11;  
        int b = 3;  
        System.out.println(a * b);  
        System.out.println(a / b);  
        System.out.println(a % b);  
        System.out.println(a != b);  
    }  
}
```

b) Hva blir skrevet ut på skjermen når denne koden blir utført:

```
public class HvaSkjer2 {  
  
    public static void main(String[] args) {  
  
        int tall = 16;  
  
        if (tall > 0 || tall < 10) {  
            System.out.println("Eple");  
        }  
        if (tall > 0 && tall < 10) {  
            System.out.println("Banan");  
        }  
        System.out.println(mystisk(tall));  
    }  
  
    private static int mystisk(int x) {  
        int tall = 3;  
        if (x > 10) {  
            tall = tall + 5;  
        }  
        return tall;  
    }  
}
```

c) Hva blir skrevet ut på skjermen når denne koden blir utført:

```
public class HvaSkjer3 {  
  
    public static void main(String[] args) {  
  
        System.out.print(1);  
        try {  
            System.out.print(2);  
            int x = 1 / 0; //Resulterer i en ArithmeticException  
            System.out.print(3);  
        } catch (ArithmeticException e) {  
            System.out.print(4);  
        }  
        System.out.print(5);  
    }  
}
```

## Oppgave 2 – 15 % (~ 36 minutter. Deloppgaver teller 5% ~ 12 minutter)

- a)
- Hva er en konstant? Gi et eksempel på deklarasjon av en konstant i Java.
  - Hvorfor bruker vi konstanter i et dataprogram?
- b)
- Hva vil det si å overlaste (*en. overload*) en metode? Gi et eksempel i Java.
  - Hva vil det si å overskrive (*en. override*) en metode? Gi et eksempel i Java.
- c)
- Forklar hva innkapsling og informasjonsskjuling er.
  - Hvilken rolle spiller adgangsmarkører (*en. access modifiers*) i denne sammenhengen?

## Oppgave 3 – 20% (~ 48 minutter. Deloppgaver teller 10% ~ 24 minutter)

- a) Vedlegg A inneholder API-dokumentasjon for en klasse TilfeldigeTall som kan brukes til å generere tilfeldige heltall i et gitt intervall. Skriv en main-metode som:
- Lar brukeren taste inn en min- og maksverdi (heretter kalt min og maks).
  - Som bytter om disse hvis inntastet min er større enn inntastet maks.
  - Som bruker klassen TilfeldigeTall til å trekke 100 tilfeldige heltall i intervallet min ... maks.
  - Beregner og skriver ut gjennomsnittsverdien på skjermen.
- b) Anta at du har en metode `public static void snuRekkefølgen(String[] tab)` som snur rekkefølgen på elementene i en tabell. Du kan anta at metoden finnes i samme klasse som main. Skriv en main-metode som:
- Oppretter en String-tabell `endimTab` med 5 hardkodete verdier (bruker tabellinitialisering, dvs. lagrer verdier direkte i tabellen når den blir opprettet).
  - Oppretter en todimensjonal 3x3 String-tabell `todimTab` med 9 hardkodete verdier.
  - Bruker metoden `snuRekkefølgen()` til å snu rekkefølgen på elementene i `endimTab`.
  - Bruker metoden `snuRekkefølgen()` til å snu rekkefølgen på elementene i hver enkelt rad i `todimTab`.
  - Skriver ut verdiene i den snudde `endimTab`, helst ved hjelp av den forenklede for-løkken.
  - Skriver ut verdiene i den snudde `todimTab`, en rad (3 verdier) per linje, helst ved hjelp av forenklede for-løkker.

**Oppgave 4 – 50%** (~ 120 minutter. a) teller 13% ~ 30 minutter, b) teller 25% ~ 60 minutter, c) teller 12% ~ 30 minutter)

I denne oppgaven skal du først lage noen klasser som inngår i et system for å administrere en samling instruksjonsvideoer. Klassene vi skal se på er: `Video` og `Videoarkiv`. Klassene er nærmere beskrevet i delspørsmålene nedenfor.

a) Lag klassen `Video`. Klassen skal inneholde:

- Feltvariabler som skal være private.
  - `tittel` – Tittel på videoen (tekststreng).
  - `produsent` – Navn på den som har produsert videoen.
  - `sekund` – Lengden på videoen i hele sekunder.
- Konstruktør med parametere der vi kan gi verdi til alle feltvariablene.
- Hent- og settmetoder for alle feltvariablene.
- `toString()` metode som returnerer en streng på formen

"Tittel: **Løkker i Java**, Produsent: **Ole Olsen**, Tid: **5.02**".

Legg merke til at tiden skal være i minutter og sekunder og at sekundene skal vises med to siffer.

b) Klassen `Videoarkiv` inneholder en samling videoer. Klassen skal inneholde:

- Feltvariabler som skal være private.
  - `vTab` – tabell som inneholder videoer.
  - `antal` – antall videoer som faktisk er i tabellen.
- Konstruktør  
`public Videoarkiv(int maksAntal)`  
Konstruktøren skal opprette en tabell med plass til `maksAntal` videoer.
- Public metoder
  - `void leggTil(Video v)`  
Instansmetode som legger til en video i neste ledige posisjon i tabellen dersom der er plass. Dersom tabellen er full skal det skrives en feilmelding.
  - `int finnTotalTid()`  
Instansmetode som returnerer samlet tid (i sekunder) for alle videoene i arkivet.
  - `void listAlle(String produsent)`  
Instansmetode som lister alle videoer av en produsent (gitt som parameter). Først skal metoden skrive en passende overskrift. Deretter en linje med bare tittel for hver video av produsenten. Til slutt skal antall videoer knyttet til produsenten skrives ut.



- o `void slett(String tittel)`  
Instansmetode som sletter en video med gitt tittel (kan anta det finnes høyst en). Dersom tittelen ikke finnes skal det skrives en passende feilmelding. Dersom tittelen finnes skal tilhørende video slettes og etter sletting skal fremdeles alle videoene ligge etter hverandre i starten av tabellen (skal ikke være "hull").

- o `void bestePar(int tidsgrense)`

Tips: Denne metoden er ganske arbeidskrevende, så om du har knapt med tid, kan det være lurt å ta deloppgave c) først.

Instansmetode som finner og skriver ut de **to** videoene som passer best innenfor en tidsgrense. Det betyr at summen av lengdene på de to videoene skal være så stor som mulig, men ikke over tidsgrensen. Metoden skal også gi en fornuftig utskrift om det ikke finnes et par med samlet tid innenfor grensen.

Eksempel (har brukt små tall så det skal være lettere å følge med)

Tider: 14, 5, 13, 9, 8

Tidsgrense: 16

Nå er det videoene med tid 5 og 9 (totaltid 14) som passer best selv om de med 9 og 8 (samlet tid 17) er nærmere tidsgrensen.

**Utskrift:** <tittelNr1> og <tittelNr2> passer best.

- c) Lag en main-metode du bruker klassene `Video` og `Videoarkiv`. Du skal opprette et arkiv med plass til 100 videoer. Legg til 3 videoer. List alle videoer produsert av "Ole Olsen". Skriv ut beste par som passer innenfor en tidsgrense på 10 minutter. Slett videoen med tittel "Løkker i Java".

Lykke til!

## Vedlegg A

[Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

[Summary](#) [Nested](#) [Field](#) [Constr](#) [Method](#) [Detail](#) [Field](#) [Constr](#) [Method](#)

### Class TilfeldigeTall

java.lang.Object  
TilfeldigeTall

---

```
public class TilfeldigeTall
extends java.lang.Object
```

En klasse som genererer tilfeldige heltall i et gitt intervall.

#### Constructor Summary

**Constructors**

Constructor and Description
TilfeldigeTall(int min, int max) Lager et TilfeldigeTall-objekt som genererer tilfeldige tall i intervallet min ... max (Både min og max er med i intervallet).

#### Method Summary

**Methods**

Modifier and Type	Method and Description
int	nesteTall() Gir neste tilfeldige tall.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### TilfeldigeTall

```
public TilfeldigeTall(int min,  
                     int max)
```

Lager et TilfeldigeTall-objekt som genererer tilfældige tall i intervallet min ... max (Både min og max er med i intervallet).

#### Parameters:

**min** - Det minste tallet i intervallet

**max** - Det største tallet i intervallet

## Method Detail

### nesteTall

```
public int nesteTall()
```

Gir neste tilfældige tall.

#### Returns:

Neste tilfældige tall i min ... max

[Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

[Summary: Nested](#) | [Field](#) | [Constr](#) | [Method](#) [Detail: Field](#) | [Constr](#) | [Method](#)