Oppgaver

Begynn gjerne med noen veldig enkle oppgaver slik at studentene blir kjent med syntaksen i Java, at all kode må tilhøre en klasse, at man trenger et hovedprogram og at man må kompilere osv. Gi gjerne noen oppgaver som har med løkker å gjøre, slik at de blir vant til hvordan en for-løkke ser ut f.eks.
Dersom studentene synes dette er for enkelt, kan du se på oppgavene for neste uke som er lagt til nedenfor.
Oppfordre dem til å jobbe med oppgavene sammen. **NB!** Oppfordre studentene til å jobbe videre med oppgaver hjemme som de ikke får til i timen.

1.a: Skriv klassen HelloWorld.java. Klassen skal inneholde en main-metode. I main-metoden skal programmet først la brukeren skrive inn en tekststreng med navnet sitt. (Hint: Bruk et Scanner-objekt for å gjøre dette - husk å importere den nødvendige klassen.) Så skal programmet skrive ut en streng som bruker navnet til brukeren.
Eks.
> Hei! Hva heter du?
> Trude
> Hei Trude! Velkommen til IN1010.

```
import java.util.Scanner;

class HelloWorld{
    public static void
main(String[] args) {
        System.out.println("Hei!
Hva heter du?");

        Scanner sc = new
Scanner(System.in);
        String name =
sc.nextLine();

        System.out.println("Hei" +
name + "Velkommen til
IN1010.");
    }
}
```

2: Det følgende programmet er skrevet i Python. Skriv om programmet til Java-syntaks. (Alle metodene i programmet skal være 'public'):

```
class Person:
        def __init__(self, alder, navn):
        self._alder = alder
        Self._navn = navn

        def skriv(self):
        print(self._navn, self._alder)

    def bursdag(self):
        self._ alder += 1
```

```
class Person {
    private int alder;
    private String navn;

    public Person(Int alder, String
navn) {
        this.alder = alder;
        This.navn = navn;
    }

    public void skriv() {
        System.out.println(this.navn +
"" + this.alder);
    }

    public void bursdag() {
        This.alder += 1;
    }
}
```

3.a: Skriv klassen Motorsykkel.java. Denne klassen skal inneholde følgende variabler:

- private int *kilometerstand*
- private String *regNr*
- private int *produksjonsnr*

Klassen skal også inneholde en konstruktør som tar imot registreringsnummeret. Instansvariabelen kilometerstand skal starte med verdien 0. Klassen skal også inneholde en private static int teller, som starter med verdien 0. Denne skal du bruke i konstruktøren slik at hvert nye motorsykkel-objekt får et unikt produksjonsnummer.

```
class Motorsykkel{
    private int
kilometerstand;
    private String regNr;
    private int
produksjonsnr;
    private static int
teller = 0;

    public
Motorsykkel(String
regNr) {

this.kilometerstand = 0;
        this.regNr =
regNr;

        this.produksjonsnr
= counter;
        counter++;
    }
}
```

3.b: Skriv en metode public int hentKilometerstand. Metoden skal ikke ta imot noen parametere, men skal returnere antall kilometer motorsykkelen har kjørt. Skriv en lignende metode public int hentProduksjonsnr.

```
public int
hentKilometerstand()
{
        return
kilometerstand;
    }

public int
hentProduksjonsnr() {
        return
produksjonsnr;
    }
```

3.c: Skriv metoden public void kjoer som tar imot et parameter int antKilometer. Metoden skal legge til antall kilometer til kilometerstanden.

```
public void kjoer(int
antKilometer) {
        kilometerstand +=
antKilometer;
    }
```

3.d:
Skriv en klasse Motorsykkelprogram.java. Klassen skal inneholde en main-metode. Opprett et objekt av typen Motorsykkel inne i main-metoden med et registreringsnr.

Skriv en while-løkke som skal kjøre 5 ganger. For hver runde i løkken, kall på motorsykkel-objektets kjoer-metode med 10 som argument

```
class Motorsykkelprogram{
   public static void
main(String[] args) {
      Motorsykkel m1 = new
Motorsykkel("AB1234");

      int t = 0;
      while (t < 5) {
         m1.kjoer(10);
         t++;
      }
   }
}
```

3.e: Hvilket resultat får du om du kaller på metoden hentKilometerstand nå?

```
System.out.println(m1.hentKilometerstand()
()); // 50
```

3.f: Hvis vi lager to Motorsykkel-objekter til, hvilke produksjonsnr vil de få?

```
Motorsykkel m2 = new
Motorsykkel("HELLO1");
      Motorsykkel m3 = new
Motorsykkel("22TUUT");
      System.out.println("m2:" +
m2.hentProduksjonsnr()); // 1
      System.out.println("m3:" +
m3.hentProduksjonsnr()); // 2
```

4.a: Hvilke feil finner du i følgende klasser?

```
class Boat {
   private String regNr;
   private int kilometer;


   public Boats (String regNr) {
      this.regNr = regNr;
      this.mileage = 0;
   }

   // Printer info om båten
   public void writeBoat() {
      print(regNr);
      print(mileage);
   }
}
```

Feil navn på konstruktøren

// Feil navn (mileage / kilometer)

// Må være System.out.println ()

4.b: Gitt følgende variabler:
`int a = 3;`
`String b = "4";`
`double c = 10.2;`
 	
Er de følgende kodesnuttene fungerende? I så fall, hva blir skrevet ut?	

`System.out.println(a + 5);`	8
`System.out.println(a + b);`	34
`int sum = a + b;` `System.out.println(sum);`	Nei
`int sum = a + c;` `System.out.println(sum);`	Nei