

1)

Vilka utskrifter blir true och vilka blir false?

```
public class Uppg1 {
    public static void main(String[] args){
        Integer a = new Integer(1);
        Integer b = new Integer(1);
        Integer c = new Integer(2);
        Integer d = b;

        System.out.println("a");
        System.out.println(a==a);
        System.out.println(a.equals(a));
        System.out.println(a==b);
        System.out.println(a.equals(b));
        System.out.println(a==c);
        System.out.println(a.equals(c));
        System.out.println(a==d);
        System.out.println(a.equals(d));

        System.out.println("\nb");
        System.out.println(b==a);
        System.out.println(b.equals(a));
        System.out.println(b==b);
        System.out.println(b.equals(b));
        System.out.println(b==c);
        System.out.println(b.equals(c));
        System.out.println(b==d);
        System.out.println(b.equals(d));

        System.out.println("\nc");
        System.out.println(c==a);
        System.out.println(c.equals(a));
        System.out.println(c==b);
        System.out.println(c.equals(b));
        System.out.println(c==c);
        System.out.println(c.equals(c));
        System.out.println(c==d);
        System.out.println(c.equals(d));

        System.out.println("\nd");
        System.out.println(d==a);
        System.out.println(d.equals(a));
```

## SI-Pass 4

[tinyurl.com/sipass4](http://tinyurl.com/sipass4)

```
        System.out.println(d==b);
        System.out.println(d.equals(b));
        System.out.println(d==c);
        System.out.println(d.equals(c));
        System.out.println(d==d);
        System.out.println(d.equals(d));
    }
}
```

e)

Vad är skillnaden på == och equals?

**2)**

Var ska variablerna a-e deklarerars för att koden ska fungera och gå igenom samtliga rader?

Observera att variablerna ska deklarerars så långt in som möjligt för att inte slösa med minne (instansvariabler är alltså INTE ok).

```
public class Uppg2 {
    public static void main(String[] args){

        for(int i=0; i<a; i++){

            while(b){

                do {

                    if(c==null){

                        d = a++;
                    }
                } while(e!=a);
            }
        }
    }
}
```

## SI-Pass 4

[tinyurl.com/sipass4](http://tinyurl.com/sipass4)

## 3)

Skriv följande metoder:

## a)

En metod som får en int-array som parameter och skriver ut alla siffror i array:en i tur och ordning.

## b)

En metod som kopierar en int-array genom att få en array som parameter och returnera en kopia av arrayen.

## c)

En metod som skriver över en int-array så att alla dess siffror ändras till 0.

## 4)

Tänk er en klass för ett monster. Den har instansvariabler för sina egenskaper som antal armar osv:

```
public class Monster {  
    private int nbrOfArms;  
    private int nbrOfLegs;  
    private int nbrOfEyes;  
    private int nbrOfNoses;  
  
    //Skriv konstruktorer här:  
  
}
```

## a)

Skriv en konstruktor med en parameter som anger antal armar på monstret som skapas.

## b)

Skriv en konstruktor med två parameter som anger antal ben och ögon på monstret som skapas.

## c)

Skriv en konstruktor med parametrar som anger alla fyra egenskaper på monstret som skapas.

## SI-Pass 4

[tinyurl.com/sipass4](http://tinyurl.com/sipass4)

d)

Diskutera vad som händer om man nu anropar defaultkonstruktorn (alltså utan parametrar):

- `new Monster();`

e)

Vad är en konstruktor till för?

f)

Hur anropas en konstruktor?

**5)**

Vi låtsas att Java inte tillåter addition av tal som är större än 1. Vill man addera med mer än 1 måste man nu addera och subtrahera med 1 i taget.

Om man till exempel vill utföra  $4+3$  så kan man göra följande:

- $(4+1)+(3-1) = 5+2$
- $(5+1)+(2-1) = 6+1$
- $(6+1)+(1-1) = 7+0$
- $7+0 = 7$

Skriv en rekursiv metod som gör operationerna ovan. Diskutera i gruppen vad rekursivt innebär och hur ni bör gå till väga. Skriv därefter koden på papper (as always).

**Överkurs)**

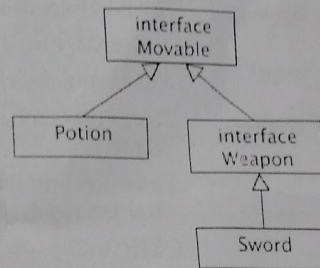
Få den rekursiva metoden att fungera även för negativa tal.

## Tentafråga från 2011-10-21)

Problem 1. Uppvärmning:

- a) Antag `int i=2; int j=3;` Vad har följande uttryck för värde? `(double) (j/i)`
- b) Antag att `i` och `j` är `int`'s större än 0, vad är det minsta värdet på `i%j`?
- c) Antag att variabeln `i` innehåller ett tal i intervallet 0..99. Skriv satser som lagrar entalsiffran i variabeln `ones` och tiotalssiffran i variabeln `tens`.  
Ex: `i=57` skall ge `ones = 7` och `tens = 5`.
- d) Antag att vi har arvshierarkin enligt figur till höger och definitionerna:

```
Movable movable;  
Weapon weapon;  
Sword sword;  
Potion potion;  
Sword klassen innehåller en metod:  
    public void sharpen()...  
Potion innehåller  
    public Potion mix(Potion  
                      other)...
```



Weapon innehåller

```
    public void break()...
```

Vilka av de följande satserna är legala? (dvs ok att göra)

- |   |  |
|---|--|
| a) <code>movable = sword;</code>        | e) <code>((Sword)movable).sharpen();</code>  |
| b) <code>sword = weapon;</code>         | f) <code>potion.mix(movable);</code>         |
| c) <code>weapon = (Sword)potion;</code> | g) <code>potion.mix((Potion)movable);</code> |
| d) <code>weapon.sharpen();</code>       | h) <code>sword.break();</code>               |