

Gjennomgang  
prøveeksamen 2014  
oppgave 1, 2, 4, 5, 7

1a: Hva er verdien til tall etter at følgende kode er utført: **17**

```
int tall = 5;  
tall = (tall*3) + 2;
```

5 15 17

1b: Anta at følgende programsetninger utføres. Hva skrives ut på skjermen?

```
int i = 11;
int j = i;
int k = 32;
if (k > j * i || k < i) {
    System.out.println("A");
} else {
    if (k < j * i && k > i){
        System.out.println("B");
    } else {
        System.out.println("C");
    }
}
```

1b) Anta at følgende programsetninger utføres. Hva skrives ut på skjermen: **B**

```
int i = 11;
int j = i;
int k = 32;
if (32 > 121 || 32 < 11) {
    System.out.println("A");
} else {
    if (32 < 121 && 32 > 11){
        System.out.println("B");
    } else {
        System.out.println("C");
    }
}
```

# 1c: Hva skrives ut her?

```
String sekv = "0";  
for (int i=0; i<6; i++) {  
    sekv = sekv + i;  
}  
System.out.println ("sekv = " + sekv);
```

1c: Hva skrives ut her:

0012345

```
String sekv = "0";  
for (int i=0; i<6; i++) { i=6  
    sekv = sekv + i; sekv=0012345  
}  
System.out.println ("sekv = " + sekv);
```

2a: Gitt følgende kode. Hva returneres fra metodekallet `rest(45,7)`?

```
int rest (int n, int m) {  
    while (n >= m) {  
        n-=m;  
    }  
    return n;  
}
```

2a: Gitt følgende kode. Hva  
returneres fra metodekallet rest  
(45,7): 3

```
int rest (int n=45, int m=7) {  
    while (n >= 7) {  
        n-=m; n=45,38,31,24,17,10,3  
    }  
    return n; n=3  
}
```



2b: Anta at følgende program utføres. Hva blir utskriften på skjermen?

```
class Studentregister {  
    public static void main (String[] args) {  
        Student s = new Student ("Ole", "Karl Johans gt 1");  
        Student p = new Student ("Marit", "Karl Johans gt 2");  
        System.out.println (s.faaNavn() + " og " + p.faaNavn());  
    }  
}
```

```
class Student {  
    private String navn = "Grete";  
    private String adresse = "Blindernveien 3";  
    public Student (String navn, String adresse) {  
        this.navn = navn;  
        this.adresse = adresse; }  
    String faaNavn() {  
        return navn; }  
}
```

2b: Anta at følgende program utføres. Hva blir utskriften på skjermen: **Ole og Marit**

```
class Studentregister {  
    public static void main (String[] args) {  
        Student s = new Student ("Ole", "Karl Johans gt 1");  
        Student p = new Student ("Marit", "Karl Johans gt 2");  
        System.out.println (s.faaNavn() + " og " + p.faaNavn());  
    }  
}
```

**Ole**                      **og**                      **Marit**

```
class Student {  
    private String navn = "Grete";  
    private String adresse = "Blindernveien 3";  
    public Student (String navn, String adresse) {  
        this.navn = navn;  
        this.adresse = adresse; }  
    String faaNavn() {  
        return navn; }  
}
```

4: Du skal nå skrive en metode som har tre parametere av typen **double** og som returnerer en verdi av typen **double**. Metoden skal finne den største av de tre parameterverdiene og returnere denne.

Hvis metoden heter **storst**, så skal f.eks. setningen

**double v = storst (0, 10.2, 4.7);**

føre til at variabelen **v** blir tilordnet verdien **10.2**

4: Du skal nå skrive en metode som har tre parametere av typen **double** og som returnerer en verdi av typen **double**. Metoden skal finne den største av de tre parameterverdiene og returnere denne.

Hvis metoden heter **storst**, så skal f.eks. setningen

**double v = storst (0, 10.2, 4.7);**

føre til at variabelen **v** blir tilordnet verdien **10.2**

```
double storst(double a, double b, double c){
    double svar=a;
    if (b > svar){
        svar = b;
    }
    if (c > svar){
        svar = c;
    }
    return(svar);
}
```

5a: Du skal skrive en metode med en **int**-array som parameter og som returnerer en **int**-array.

Metoden skal opprette en ny **int**-array som er dobbelt så lang som den i parameteren, kopiere over verdiene i parameter-arrayen i første halvdel av den nye arrayen (og la de resterende verdiene i den nye arrayen være 0), og til slutt returnere den nye arrayen.

5a: Du skal skrive en metode med en **int**-array som parameter og som returnerer en **int**-array.

Metoden skal opprette en ny **int**-array som er dobbelt så lang som den i parameteren, kopiere over verdiene i parameter-arrayen i første halvdel av den nye arrayen (og la de resterende verdiene i den nye arrayen være 0), og til slutt returnere den nye arrayen.

```
int[] Oppgave5a(int[] a){
    int[] b = new int[a.length*2];
    for (int i = 0; i<a.length; i++) {
        b[i] = a[i];
    }
    return b;
}
```

5b: [..] Mer presist skal du lage en metode som har to parametre: en String-array og en String. Metoden skal konkatenerere hvert enkelt element i String-arrayen med den angitte String-verdien og returnere resultatet i en ny String-array som er like lang som den opprinnelige. Hvis metoden heter **konkatener**, så skal f.eks.

```
String[] ordliste = {"Karoline", "Anders", "Camilla"};
```

```
String[] resultat = konkatener(ordliste, " er flink");
```

resultere i at **resultat** blir en String-array av lengde 3 med verdiene "Karoline er flink", "Anders er flink" og "Camilla er flink".

5b: [..] Mer presist skal du lage en metode som har to parametre: en String-array og en String. Metoden skal konkatenerere hvert enkelt element i String-arrayen med den angitte String-verdien og returnere resultatet i en ny String-array som er like lang som den opprinnelige. Hvis metoden heter **konkatener**, så skal f.eks.

```
String[] ordliste = {"Karoline", "Anders", "Camilla"};
```

```
String[] resultat = konkatener(ordliste, " er flink");
```

resultere i at **resultat** blir en String-array av lengde 3 med verdiene "Karoline er flink", "Anders er flink" og "Camilla er flink".

```
String[] konkatener(String[] liste, String tillegg){  
    String[] utvidet = new String[liste.length];  
    for (int i=0; i<liste.length; i++){  
        utvidet[i] = liste[i] + tillegg;  
    }  
    return utvidet;  
}
```



7: Skriv en metode med en tekststreng som parameter. Metoden skal skrive en rekke linjer ut på skjermen, etter følgende mønster: Første linje inneholder teksten i parameteren. Linje to inneholder samme tekst som første linje - men med første linjes første tegn som siste tegn på linje to, og resten av tegnene flyttet en posisjon frem. Dette gjentas inntil første tegn på den siste linjen er det som var siste tegnet på den første linjen. Du stopper altså utskriften rett før du ville skrevet ut en linje som er maken til den første linjen.

7: Skriv en metode med en tekststreng som parameter. Metoden skal skrive en rekke linjer ut på skjermen, etter følgende mønster: Første linje inneholder teksten i parameteren. Linje to inneholder samme tekst som første linje - men med første linjes første tegn som siste tegn på linje to, og resten av tegnene flyttet en posisjon frem. Dette gjentas inntil første tegn på den siste linjen er det som var siste tegnet på den første linjen. Du stopper altså utskriften rett før du ville skrevet ut en linje som er maken til den første linjen.

```
static void rotasjoner(String tekst){
    for (int splitt=0; splitt<tekst.length(); splitt++){
        System.out.println(tekst.substring(splitt,tekst.length())
            + tekst.substring(0,splitt));
    }
}
```

7: Skriv en metode med en tekststreng som parameter. Metoden skal skrive en rekke linjer ut på skjermen, etter følgende mønster: Første linje inneholder teksten i parameteren. Linje to inneholder samme tekst som første linje - men med første linjes første tegn som siste tegn på linje to, og resten av tegnene flyttet en posisjon frem. Dette gjentas inntil første tegn på den siste linjen er det som var siste tegnet på den første linjen. Du stopper altså utskriften rett før du ville skrevet ut en linje som er maken til den første linjen.

```
static void rotasjoner(String tekst){
    for (int iterasjon=0; iterasjon<tekst.length(); iterasjon++){
        System.out.println(tekst);
        tekst = tekst.substring(1,tekst.length())
                + tekst.substring(0,1);
    }
}
```