

Oppgave 1

a)

Hva er definisjonen på teksten under?

"Lykke til med oppgavene!"

Variabel	Verdi
Klasse	Metode

Svar:

Definisjonen man er ute etter her er verdi, fordi det er noe innenfor to "anførselstegn". Det er ikke variabel fordi det deklarerer ingen variabler, og det er ikke metode for det er ikke en gyldig metodesignatur. Derimot *kan* Klasse være riktig, iogmed at "Lykke til med oppgavene!" er en *instans* av type **String** (som er en klasse). Riktignok er det *ikke* definisjonen på en klasse, da det ikke er angitt **class**. Så det spørs hvordan man ser på Klasse i svaret.

b)

Hva er verdien i variabelen `setning` etter at koden under er eksekvert?

```
String setning = "tidlig start";
System.out.println("Gry" + setning);
```

Svar:

Verdien i variabelen `'setning'` forblir uendret, og er derfor `"tidlig start"`. Dette er et typisk eksamensspørsmål, der det som skjer *i terminalen* er annerledes fra hva oppgaveteksten spør etter. På terminalen vil det komme ut `Grytidlig start`, men det er ikke det oppgaven spør etter.

c)

Hva er verdien i variabelen `cashString` etter at koden under er eksekvert?

```
String cashMoney = "Inntekt paa $1000";
String cashString = "Ingen inntekt.";

if (cashMoney.charAt(0) == '$') {
    cashString = cashMoney;
}
```

Svar:

Verdien i 'cashString' forblir uendret som "Ingen inntekt." fordi if-testen sjekker kun om char på posisjon 0 er '\$' i String-verdien "Inntekt paa \$1000" - men char verdien på posisjon 0 er 'I', som gjør at koden inni if-blokken *ikke* blir kjørt og endrer derfor ikke verdien i variabelen 'cashString'.

d)

Hva er verdien i variabelen langSetning etter at koden under er eksekvert?

```
String setning = "Snart er det jul.";
String langSetning = "Og jula varer helt til påske.";
setning = setning + " " + langSetning;
langSetning += " Nei det gjør den ikke."
```

1. "Snart er det jul."
2. "Nei det gjør den ikke."
3. "Snart er det jul. Og jula varer helt til påske. Nei det gjør den ikke"
4. "Og jula varer helt til påske. Nei det gjør den ikke"

Forklaring:

Når koden er ferdig vil verdien være 4. "Og jula varer helt til påske. Nei det gjør den ikke." Forklaring:

```
String setning = "Snart er det jul.";
// Lagrer verdien "Snart er det jul." i variabelen 'setning'
String langSetning = "Og jula varer helt til påske.";
// Lagrer verdien "Og jula varer helt til påske." i variabelen
// 'langSetning'
setning = setning + " " + langSetning;
// Tar verdien fra 'setning', legger til " " og verdien fra
// 'langSetning' (uten at den endres noe), og lagrer det i
// variabelen 'setning', som blir "Snart er det jul. Og
// jula varer helt til påske."
langSetning += " Nei det gjør den ikke."
// Tar verdien i 'langSetning' (grunnet +=) og legger til verdien "
// Nei det gjør den ikke." og lagrer resultatet i
// variabelen 'langSetning' som blir "Og jula varer helt
// til påske. Nei det gjør den ikke." som er det oppgaven
// spør etter..
```

e)

Hvor lang vil *length()*-metoden si at variabelen `lengde` er under?

```
String lengde = "12";
System.out.println(lengde.length());;
```

Svar:

Lengden på strengen er 2, fordi det er to tegn (`char` verdier) i stringen. Det har ingenting å si *hva* tegnene er.

f) (utfordring)

Hva er verdien i variabelen `cashString` etter at koden under er eksekvert?

```
String cashMoney = "Inntekt paa $1000";
String cashString = "Ingen inntekt.";

for (int i=0; i<cashMoney.length(); i++) {
    if (cashMoney.charAt(i) == '$') {
        cashString = cashMoney.substring(i);
        i = cashMoney.length();
    }
}
```

Svar:

Verdien i variabelen '`cashString`' vil være "\$1000" etter kjøring. Det kodesnutten gjør er å sjekk hver enkelt posisjon sin *char* i verdien til '`cashMoney`' etter tegnet '\$'. Hvis tegnet finnes lagrer den en substring fra *samme* posisjon som tegnet og ut resten av strengen, før den setter '`i`' lik lengden for å avslutte for-løkken.

Oppgave 2

Skriv et program som spør brukeren om informasjonen *navn*, *telefonnummer*, *alder*, *gateadresse* og *postnummer* som skal lagres i typene *String*, *String*, *int*, *String*, *int*. Fokuser på god utskriftskultur!

Forventet løsning:

```

import java.util.Scanner;

public class Oppgave2 {

    public void metode() {
        String navn, telefonnummer, gateadresse;
        int alder, postnummer;

        Scanner stdin = new Scanner(System.in);

        System.out.println("Skriv ditt navn:");
        navn = stdin.nextLine();

        System.out.println("Skriv ditt telefonnummer (paa det
format du vil)");
        telefonnummer = stdin.nextLine();

        System.out.println("Hvor gammel er du? (skriv et
tall)");
        alder = Integer.parseInt(stdin.nextLine());

        System.out.println("Hva er gateadressen din?");
        gateadresse = stdin.nextLine();

        System.out.println("Til slutt, postnummeret som et
tall");
        postnummer = Integer.parseInt(stdin.nextLine());

    }

}

```

Bedre løsning:

```

import java.util.Scanner;

public class Oppgave2Better {

    Scanner stdin = new Scanner(System.in);

    public void metode() {
        String navn, telefonnummer, gateadresse;
        int alder, postnummer;

        navn = fraBruker("Skriv ditt navn");

        telefonnummer = fraBruker("Skriv ditt telefonnummer
(paa det format du vil)");

        alder = Integer.parseInt(fraBruker("Hvor gammel er du?
(skriv et tall)"));

        gateadresse = fraBruker("Hva er gateadressen din?");

        postnummer = Integer.parseInt(fraBruker("Til slutt,
postnummeret som et tall"));

    }

}

```

```

    }

    private String fraBruker( String forklaring ) {
        System.out.print(forklaring + ": ");
        return stdin.nextLine();
    }
}

```

Oppgave 3

Klarer du å se resultatet av følgende kode?

(Hint: Sjekk *String.java* sin Java Doc¹ for hva *String.valueOf()*-metoden gjør)

```

class Student {
    String navn, telefonnummer, gateadresse;
    int alder, postnummer;

    public Student(..) {
        this.navn = navn;
        this.telefonnummer = telefonnummer;
        this.alder = alder;
        this.gateadresse = gateadresse;
        this.postnummer = postnummer;
    }

    public String hentInfo() {
        String result = "Studenten '" + navn;
        result += "' er " + String.valueOf(alder) +
            " aar gammel.";
        result += " Studenten har telefonnummer: " +
            telefonnummer;
        result += ". Og bor paa " + gateadresse +
            " " + String.valueOf(postnummer);
        return result;
    }
}

```

```

class StudentMain {
    public static void main(String[] args) {
        Student per = new Student("Per",          // navn
                                   "44 00 13 13",  // telefonnummer
                                   19,              // alder
                                   "Vestenfor 13", // gateadresse
                                   1366);           // postnummer
        System.out.println(per.hentInfo());
    }
}

```

¹<http://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

Notis:

Som forklart i gjennomgangen er ikke `String.valueOf(..)` nødvendig for resultatet - da de to neste linjene er ekvivalente:

```
result += "' er " + String.valueOf(alder) +  
          " aar gammel.";
```

er det samme som

```
result += "' er " + alder +  
          " aar gammel.";
```

Det ble inkludert fordi man må ha metoden hvis man vil lagre noe annet enn `String` som `String` uten å legge til en tom `String`, altså:

```
String alderTekst = String.valueOf(alder);
```

er det samme som

```
String alderTekst = alder + "";
```