

# Demo 3 / 29.9

## Tasks

1. **Information presentation on a computer:** Let us presume that a computer has only 8 bit numbers. So that is the space a result must fit in. Add up (one on top of the other) the following binary numbers carefully. Remember to write down the carry digits (also the one that doesn't fit on the left flank):

a)	b)	c)	d)
0101 1111	0101 1111	0000 0001	1111 1111
0001 1111	0101 1111	1111 1111	1111 1111

In modern computers it's common to present negative numbers using what is called the two's complement. This means that a positive number is made negative by reversing all the bits and adding 1 to the result. In turn, when the number is converted into "human form" the first bit is checked and if it is 1, the number is negative. You can then make it positive again by taking two's complement again (all bits reversed and +1). In such a case, when printing a - character is first printed and then the value of the number. For example:

3 = 0000 0011

-3 is as follows:

1) all reversed	1111 1100
2) +1	= 1111 1101 = -3

In the same way if we have the binary numbers 0010 1101 and 1101 1111 and we wanted to decode them, the decoding should start out as follows:

0010 1101 the number is positive, or 45  
 1101 1111 the number is negative, so first two's complement  
 0010 0000 + 1 = 0010 0001 = 33, so the result is -33

Decode the numbers and their sums in subtasks a)-d) :

- e) when an 8 bit computer uses only positive numbers.
  - f) when an 8 bit computer uses two's complement for negative numbers.
  - g) Which results were correct in subtask e).
  - h) Which results were correct in subtask f).
  - i) Give an example where the sum of two negative numbers was positive in an 8 bit computer that uses two's complement for negative numbers.
  - j) Make a rule according to the numbers you calculated and the "false" results and carry numbers that states when you can decide a result is correct and when it is not.
2. **Variables and types:** How would you present (type and name) the variables for the following situations. Also give an example how you would set a value for a specific variable. You can return the answer as a Java file if you want.
- a) The length of your way to work by bike
  - b) The number of credits earned
  - c) The full extent of a [tutkinto] in credits
  - d) The number of demo tasks done
  - e) The first letter of a book
  - f) The time of the winner of a 100m sprint

3. **Variables and types:** Review the following variable types and names. Are they syntactically correct? Are they good choices:

```
int priceOfStrawberriesPerKilo;
double first,second,final;
char myName;
double meterMm;
double hoursInADay;
real speed;
double DistanceToTheSun;
int midTemperatureOfTheMonth;
int cats,2_chickens,threeDogs;
int i,j,l,I;
```

4. **Subroutines:** Search in the Internet how to convert Fahrenheits to Celcius degrees. Write a (function)subroutine which can be called as follows:

```
double temperatureC;
temperatureC = fahrenheitToCelsius(13);
```

Write similar call to a (function)subroutine which converts pounds (lb) to gramms. Write the (function)subroutine too. Don't forget to give proper comments as seen in example programs!

5. **Subroutines:** Download the snowman program:

<http://users.jyu.fi/~mapohjol/opetus/ohjl/tutorials/t2/Snowmen2.java>

and modify it so that there are four different ways to draw a snowman. Note that two of them already exists. Remember proper comments!

- without giving the size of any snowball
- by giving the size of the lowest snowball
- by giving the size of the two lowest snowballs
- by giving the size of every three snowballs

The program you continued in the computer lab tutorial 2 is at your disposal; you might remember the task of giving eyes and a nose to a snowman. Don't forget to "test" all the four different subroutines!

6. **Expressions:** Write expressions which calculate:

- number of seconds in a day
- number of hours in a year
- how long does it take for sunlight to reach Earth  
(mean distances are enough)
- how many kilometers is one lightyear

Note: do the task as a Java program. You don't need to create subprograms, main is enough

- B1. Create a program that asks user a string (use the class `Scanner`). Next, the string user just gave is printed first all characters in uppercase and then all characters in lowercase (look the documents of the class `String`). Finally, print first three characters of the string.

## **GURU-tasks**

G1-2 Download the snowman program from the task 5 and modify it to an object. Put 3 snowmen on the screen and make each of them to rotate in their own pace. Further information on the automated rotating can be found here: <sample/UkkoSample.java>