© Manfred Kerber
Ian Batten
Joseph Gardiner

School of Computer Science
University of Birmingham

Worksheet 1

MSc/ICY Software Workshop

Non-Assessed Exercise. Hand in: Thursday 9 October 2014 2pm.
Follow the submission guidelines on
`http://www.cs.bham.ac.uk/internal/courses/java/msc/submission.php`.

Exercise 1: (Basic) The area A of the circle is computed by $\pi \cdot r^2$. Write a `static` method `circleArea` that makes this computation, that is, that takes the variable r as parameter and return the area of the circle. Document your program appropriately and test it for values 0, 1, 2, 3, 4.4, and 5. (Hint: Use `Math.PI`.)

Exercise 2: (Basic) Between different units for measuring weight there are the following conversions:

1 ton	2240 pounds
1 hundredweight	112 pounds
1 quarter	28 pounds
1 stone	14 pounds
1 ounce	1/16 pounds
1 drachm	1/256 pounds
1 grain	1/7000 pounds
1 pound	0.45359237 kilograms

(a) Write corresponding `static` methods `ton2Kilogram`, `hundredweight2Kilogram`, `quarter2Kilogram`, `stone2Kilogram`, `ounce2Kilogram`, `drachm2Kilogram`, `grain2Kilogram`, and `pound2Kilogram`. Make use of appropriate variables. Document your program appropriately and test it for suitable values.

(b) Let a person's weight be given by two `int` values, `stones` and `pounds`. Write a `static` method `imperial2Metric` that takes these two values as parameters and returns the weight in kilograms as an `int` rounded to the next integer value. Document and test your program appropriately.

Exercise 3: (Medium) If a capital sum of pounds `capitalSum` is invested at a fixed interest rate `interestRate` then the interest is added to the capital sum at the end of each year. The `capitalSum` after `years` many years is computed according to

$$\texttt{capitalSum} * (1 + 0.01 * \texttt{interestRate})^{\texttt{years}}$$

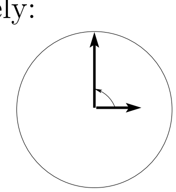
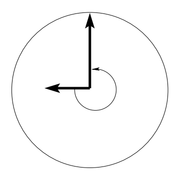
Write a corresponding `static` method `interestAddedCapitalSum` with the three parameters `double capitalSum`, `double interestRate`, and `int years`. Document and test your program appropriately.

Exercise 4: **(Advanced)** In the following `n1` and `d1` represent the numerator and denominator of the fraction $\frac{n_1}{d_1}$ and correspondingly `n2` and `d2` those of the fraction $\frac{n_2}{d_2}$, where `n1` and `n2` are integers and `d1` and `d2` positive integers. Write `static` methods which compute numbers `ns`, `ds` and `np`, `dp`, which stand for the sum and the product of the two fractions, respectively. (E.g., $\frac{1}{2}+\frac{1}{3}=\frac{5}{6}$, and $\frac{1}{2}*\frac{1}{3}=\frac{1}{6}$.)
Note that addition and multiplication on fractions are defined by:

$$\frac{n_1}{d_1}+\frac{n_2}{d_2}=\frac{n_1*d_2+n_2*d_1}{d_1*d_2} \qquad \frac{n_1}{d_1}*\frac{n_2}{d_2}=\frac{n_1*n_2}{d_1*d_2}$$

Document and test your program appropriately.

Exercise 5: **(Advanced)**
The time is 11:49. We represent the time by two variables `hours` and `minutes`, for instance, `hours = 11;` and `minutes = 49;`. Write a `static` method that takes as parameters the `hours` and `minutes` and returns the angle between the hour hand and the minute hand on a traditional analogue clock (all of type `int`). Angles should be measured counterclockwise from hour to minute hand. The result should be rounded and normalised so that it is between 0 and 359 (inclusively). For instance the angles at 3:00 and 9:00 hours should be 90° and 270°, respectively:

(Hint: 1 minute $\simeq$ 6°, 1 hour $\simeq$ 30°. Start from 12 o'clock.)
Note that your program must be able to compute the output for arbitrary inputs, in which hours may take values from 0 through 24 and minutes from 0 to 60. Do not forget to write comments which explain why your program is correct.
Document your program appropriately. Test your program for the following times: 9:00, 3:00, 18:00, 1:00, 2:30, and 4:41 (with results 270°, 90°, 180°, 30°, 255°, and 254°, respectively).