© Manfred Kerber
Alexandros Evangelidis

School of Computer Science
University of Birmingham
1 October 2019

Worksheet 1

MSc/ICY Software Workshop

Non-Assessed Worksheet

Submission Deadline is Tuesday, 8 Oct 2019, at 12:00 noon via Canvas.

Follow the submissions guidelines on Canvas. JavaDoc comments are mandatory.

Exercise 1: (Basic, 30%) The area A of a circle is computed by π times the **`radius`** squared. Write a Java method **`public static double areaCircle(double radius)`** that takes in as argument the **`radius`** and returns the area of the corresponding circle. Furthermore write a main method and try out your method for the values 0, 5, and 10.
(Hint: Use **`Math.PI`** and compute **`radius`**^{2} as **`radius * radius`**.)

Exercise 2: (Medium, 30%)

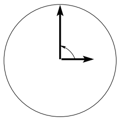
(a) Write a Java method **`public static double imperialToKg(double ton, double hundredweight, double quarter, double stone, double pound, double ounce, double drachm, double grain)`** that converts masses given in the imperial system into kilograms. Make use of the following conversions:

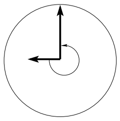
1 ton	2240 pounds
1 hundredweight	112 pounds
1 quarter	28 pounds
1 stone	14 pounds
1 ounce	1/16 pounds
1 drachm	1/256 pounds
1 grain	1/7000 pounds
1 pound	0.45359237 kilograms

(b) Write a main method and test your program by computing a person's weight in kg corresponding to 11 stones and 6 pounds. (The result should be approximately 72.5747792 kg.)

Exercise 3: (Advanced, 30%)

(a) Assume we represent the time such as 11:49 by two variables **`hours`** and **`minutes`**, that is, **`hours = 11;`** and **`minutes = 49;`**. Write a Java method **`public static int timeToAngle(int hours, int minutes)`** that computes the angle between the hour hand and the minute hand on a traditional analogue clock. Angles should be measured counterclockwise from hour to minute hand. The result should be rounded and normalized so that it is an **`int`** 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 $\simeq 6^\circ$, 1 hour $\simeq 30^\circ$. 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. Furthermore, note that the hour hand moves also when the minute hand moves, for instance, at 0:20 the hour hand will have moved by 10°. Do not forget to write comments which explain why your program is correct. Use only concepts introduced in the lecture so far (that is, no loops and no conditionals).

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 255°, respectively).

(b) Extend your program so that a third variable **seconds** takes the corresponding seconds. The program should still compute the angle between the hour hand and the minute hand. Test it for 0:00:20 and 13:05:27.272727272727 (with results 358°and 0°, respectively).

Exercise 4: (Debugging, 10%) You have the task to evaluate the following piece of code which has problems. Submit the improved code with an assessment of the original code as a comment at the start.

```
public class Swap {
    /**
     *  swaps i and j
     *  @param i 1st variable.
     *  @param j 2nd variable.
     */
    public static void
        swap(int i, int j) {
          i = j;
          j = i;
    }
```

```
 public static void main(String[] args) {
     int i = 2;
     int k = 6;
     System.out.println("Original       i: "
                        + i + " k: " + k);
     swap(i,k);
     System.out.println("After swapping i: "
                        + i + " k: " + k);
 }
}
```