**CSCM41 January 2017**
**Programming in Java**

*(Attempt 2 questions out of 3)*

## Question 1

(a) Consider the following program for teaching fractions:

```
class Q1a {
  public static void main(final String[] args) {
    final int a = Integer.parseInt(args[0]);
    final int b = Integer.parseInt(args[1]);
    System.out.println("a/b = " +  a/b);
  }
}
```

One wants to see for example

```
> java Q1a 1 2
a/b = 0.5
```

(i) But what is actually printed out for these inputs? State the output, and explain it precisely. **[3 marks]**

(ii) The printout is different from the expected result. How could you change the `println`-statement (Line 3 in `main`), so that the expected result is obtained? Explain your reasoning. **[3 marks]**

(b) Write a **function** `sum_min_max`, which takes three (single) integers as arguments, and returns the sum of their minimum and maximum. For example inputs $1, 2, 3$ and $5, -1, 2$ yield 4 (namely $1 + 3 = -1 + 5 = 4$), and $1, 1, 1$ yields 2. Take care to have a correct function-signature (the "header line"). For example, `System.out.println(sum_min_max(1,2,3));` outputs the result for the first example. **[8 marks]**

(c) Discuss when to use the access specifier `private` and when to use `public` for data members (instance variables) of a class. Explain what happens when you access a private instance variable from another class. If an error should occur, specify when this error will happen, and explain the underlying motivation for this error. Give one concrete and meaningful example, where using `public` instead of `private` can lead to serious problems.
**[5 marks]**

(d) Define a class `Point3d`, which has three `double` instance variables $x, y, z$. You need only to provide one constructor, and the method `equals`, which determines in the natural way whether two points are equal or not.
**[6 marks]**

**Question 2**

(a) Give an example of a complete Java program which reads a string from the command-line and prints "Hello, X!", where $X$ is the value of that string. **[3 marks]**

(b) Consider the simple class `X`, which has one `int` instance variable, with a constructor setting this variable, one `get`-method to obtain its value, and one `set`-method to set its value. Consider the code

```
final X x = new X(11);
System.out.println(x.get());
x.set(7);
```

(i) The code compiles — explain, why this is the case, despite the `final`. **[3 marks]**

(ii) Add a line of code, which would compile without the `final` in the `x`-definition, but does not compile with the code as above. Explain why. **[3 marks]**

(c) Arrays and loops

(i) Consider the following function:

```
private static boolean unknown(int[] A) {
  for (int i = 0; i <= A.length-1; ++i)
    if (A[i] < 0) return true;
  return false;
}
```

(ii) What is this function computing (that is, when will the result be `true`, when `false`)? Your answer should include the output in case $A$ is empty (has length zero). **[3 marks]**

(iii) Under which circumstances will `unknown` fail (an error occurs, and an exception is raised)? Which error occurs precisely, and what is the reason for this error? **[3 marks]**

(iv) Rewrite the function `unknown`, improving the coding standard, and correcting the error, so that the implementation shall now work under all circumstances, while still fulfilling the specification as you worked out under (ii). **[4 marks]**

(d) Explain the difference between a static function (static method) and a non-static function (instance method) in a class. Construct some class, and give an example of a reasonable static function for it, and an example for an instance method. **[6 marks]**

**Question 3**

(a) Loops, functions and programs

(i) Write a **function** `sorted` which takes an integer array `A` and returns `true` if the array is sorted in ascending order, otherwise `false`. For full marks, the function must return an appropriate `boolean` under all circumstances (must *never* raise/throw an exception). **[6 marks]**

(ii) Write a complete Java program which reads integers $x_1, \ldots, x_n$, $n \geq 0$, from the command line, and which outputs `true` in case the values are sorted in ascending order, and `false` otherwise, **using** the above function `sorted` (assume here that the function exists, whether you could answer the first part or not). Ignore possible wrong command-line inputs. **[4 marks]**

(b) Classes

(i) Write a class `Record`,

- which contains a string and an integer,
- which can be constructed from a string and an integer,
- which can also be constructed without arguments,
- where we can obtain the data via methods `name` and `id`,
- and where finally we can also change the data.

**[10 marks]**

(ii) Write some code which uses all constructors and methods of `Record`. **[5 marks]**