Prof. Dr. Jasmin Blanchette
Dr. Martin Desharnais-Schäfer
Dr. Michael Kirsten

Ludwig-Maximilians-Universität München
Institut für Informatik
Discussion on 28.10.2026
Homework due on 04.11.2026 at 14:15

Exercise Sheet 3 in
Scientific and Technical English
WiSe 2026/27

The exercise sheets consist of in-class exercises and homework. The in-class exercises take place during the second half of the lecture time slots. The homework, which is optional and ungraded, can be submitted via the "Homework" section in Moodle. The homework is subject to peer review.

Unless indicated otherwise, generative artificial intelligence assistants such as ChatGPT may be used, as long as you acknowledge how you use them as specified by the Institute's policy on plagiarism.[1] However, you may not use such tools to generate peer reviews for you. In addition, we strongly recommend that you do not use them to generate entire solutions, since that would defeat the purpose of the exercises.

In-class exercise 3-1 ***Transatlantic Medley*** The following text inconsistently mixes the rules of American and British English. Identify as many constructs as possible that are specific to either language variant.

> "Alex, we have a small problem", said the woman. Her irritated tone did not match the way she casually leant against the filing cabinet. "Is that a check mark?"
>
> Alex tried to analyse the splotches on the page. "Looks like an exclamation point to me."
>
> "Bloody hell. A pipe broke in the archive room, and now the writing on these is entirely unrecognisable."
>
> 'Actually, that might be more of a midsize problem then, Dr Arbour. If it's unfeasible to discern the grades, we're obligated to repeat the exam within the autumn term.' He was sceptical they would be able to organise that in time, but that was a moot point just now.
>
> Dr. Arbour frowned, and dropped the papers she was holding into the rubbish bin. "Have the committee ruled on that yet?"

[1] `https://www.medien.ifi.lmu.de/lehre/Plagiate-IfI.pdf`

"Not yet. They tabled it until after the curricula are dealt with. But you know how the maths department's vice president feels about issues of this caliber. I doubt they'll rule in our favour."

"We could always randomize the marks," suggested Dr Arbour.

"If we do that, I'll need to go update my CV." A stunt like that was sure to get his teaching licence revoked.

In-class exercise 3-2 ***Said in So Many Words*** The following text explains the concept of tail recursion.

Within the discipline of functional programming, in order for a function to be described as tail recursive, it must fulfill the following criterion: Whenever a function is called recursively, whatever result this recursive call yields must already constitute the final result of the function itself, without any additional computations occurring that further modify this recursive result.

Compared with the naive approach, such an implementation can be said to reduce the memory space a function requires for its execution: Because of the fact that the values of intermediate recursive calls do not need to be processed further, it is not necessary for the compiler to store every return call location up until the point where the entire recursion terminates.

It should, however, also be noted that this constraint on compiler optimization is not always equally applicable to all programming languages. For instance, if we consider the language Haskell as an example, we find that because evaluations are performed lazily, these considerations become redundant in any and all cases where the evaluation of the recursive call can be delayed up until such a point where its result is actually needed in the computation.

a) Rewrite the above text to make it as concise as you can.

b) Compare your version with the original version in terms of word count.

c) Which version do you find better—e.g., more easily comprehensible, more engaging, or more pleasing to read?

d) Are there further improvements you could make to your rewritten version?

Homework 3-3 ***Utterly Unstylish*** As a part of a seminar paper, you are asked to write a brief introduction to inheritance in Java. For some reason, you have decided that it is your personal objective to hand in the most poorly written text possible for this assignment. Refer back to the style principles from the lecture, and choose any number of them to focus on. You can either try to break one of them as comprehensively as possible or combine multiple avenues of stylistic offense. Choose whichever option seems the more atrocious to you. Your text should be about 250 words long.

Homework 3-4 ***From Casual to Formal*** Consider the following excerpts from two computer science texts. They are written in a rather casual register. Rewrite them to suit the level of formality appropriate for a university paper.

a) From Joshua Bloch and Neal Gafter, *Java Puzzlers: Traps, Pitfalls, and Corner Cases*, Addison-Wesley Professional, 2005:

> The behavior of this program is platform independent: It won't compile on any platform. If you tried to compile it, you got an error message that looks something like this:
>
> ```
> LinePrinter.java:3: ';' expected
> // Note: \u000A is Unicode representation of linefeed (LF)
> ^
> 1 error
> ```
>
> If you are like most people, this message did not help to clarify matters.

b) From Miran Lipovača, "Learn You a Haskell for Great Good!":[2]

> It would seem as though we've gained the ability to temporarily extract things from Maybe values without having to check if the Maybe values are Just values or Nothing values at every step. How cool! If any of the values that we try to extract from are Nothing, the whole do expression will result in a Nothing. We're yanking out their (possibly existing) values and letting >>= worry about the context that comes with those values. It's important to remember that do expressions are just different syntax for chaining monadic values.

[2] `https://learnyouahaskell.com/`