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

Ludwig-Maximilians-Universität München
Institut für Informatik
Discussion on 21.01.2026
Homework due on 28.01.2026 at 16:00

Possible solution for Exercise Sheet 13 in

Scientific and Technical English for Computer Scientists

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.

Homework 13-3 ***Getting Started with Software*** Choose some familiar piece of software (e.g., your favorite Python library). Write a brief getting-started guide of at most 300 words where you explain how to install the software. Include a description of how to create a "hello world" example and, if the word count allows for it, some slightly more complicated examples. Use pictures (e.g., screenshots, diagrams) where appropriate.

POSSIBLE SOLUTION:

The `pgfornament` package lets you draw "Vectorian" (a portmanteau of *vector* and *Victorian*) ornaments using `pgf` and `tikz`.

Installation

Use your system's LaTeX package manager (e.g., TeX Live or MiKTeX) to install the `pgfornament` package. You might need to install its dependency `tikzrput` first. If you use Overleaf, you do not need to install anything.

Your First Ornament

To load the package, simply place the following line in the preamble of your document:

```
\usepackage{pgfornament}
```

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

To insert an ornament into your document, use the `pgfornaments` macro. The command

```
\pgfornament[width=6cm,symmetry=h]{46}
```

yields the result

You might have noticed that the ornament is addressed with the hard-coded number 46. A library showcasing the various ornaments and their respective numbers can be found in the package documentation: `https://ftp.gwdg.de/pub/ctan/macros/latex/contrib/tkz/pgfornament/doc/ornaments.pdf`.

Changing Options

To customize your ornaments, you can change various options. In the first example, we have already seen usages of the options `width` and `height` to adjust the size of the ornament. Further options include `color`, `symmetry`, and `opacity`. For example,

```
\definecolor{MidnightBlue}{rgb}{0.0, 0.2, 0.4}
\pgfornament[color=MidnightBlue,width=3cm,opacity=0.5]{24}
```

generates a slightly transparent blue ornament:

Complicated Shapes

To create more sophisticated shapes, the ornaments can be used as elements (nodes or edges) of a `tikz` drawing. The `pgfornament` package automatically imports `tikz`.

To place an ornament into a node, simply add it like any other content:

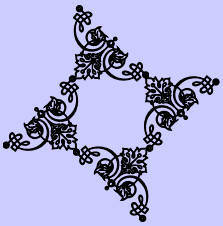
```
\begin{tikzpicture}
  \node{\pgfornament[width=1cm]{63}};
\end{tikzpicture}
```

This can be used to create more complicated shapes by using `tikz`'s many capabilities. For example,

```
\begin{tikzpicture}
  \foreach \a in {0,90,180,270}
```

```
  \node[anchor=west,rotate=\a,inner sep=0pt,xshift=12pt] {%
    \pgfornament[width=1cm]{63}};
\end{tikzpicture}
```

creates



When the `pgfornament` package is loaded, `tikz`'s `to` command supports the `ornament=` option that can be used to add ornaments to edges. For example,

```
\begin{tikzpicture}
  \node (A) at (0,0) {};
  \node (B) at (5,5) {};
  \draw [fill=MidnightBlue!30] (A) circle (2pt)
    (B) circle (2pt);
  \draw [MidnightBlue!80] (A) to [ornament=83] (B);
\end{tikzpicture}
```

creates