Writing Articles for The Art, Science, and Engineering of Programming

Tobias Pape[a], Cristina V. Lopes[b], and Robert Hirschfeld[a]

a Hasso Plattner Institute, University of Potsdam, Germany
b University of California, Irvine, USA

Abstract The Art, Science, and Engineering of Programming is a new journal created with the goal of placing the wonderful art of programming in the map of scholarly works. Many academic journals and conferences exist that publish research related to programming, starting with programming languages, software engineering, and expanding to the whole Computer Science field. Yet, many of us feel that, as the field of Computer Science expanded, programming, in itself, has been neglected to a secondary role not worthy of scholarly attention. That is a serious gap, as much of the progress in Computer Science lies on the basis of computer programs, the people who write them, and the concepts and tools available to them to express computational tasks.

The Art, Science, and Engineering of Programming aims at closing this gap by focusing primarily on programming: the art itself (programming styles, pearls, models, languages), the emerging science of understanding what works and what doesn't work in general and in specific contexts, as well as more established engineering and mathematical perspectives.

This is an example of and a guide to writing articles for The Art, Science, and Engineering of Programming.

ACM CCS 2012

- *General and reference → Computing standards, RFCs and guidelines;*
- **Applied computing → Publishing;**

Keywords programming journal, paper formatting, submission preparation

The Art, Science, and Engineering of Programming

Perspective The Art of Programming

Area of Submission Social Coding

1 A Guided Tour

The content of a paper is its most important ingredient. That being said, papers for The Art, Science, and Engineering of Programming should adhere to a common style, which is achieved with the programming LaTeX class. You are to use this class as outlined here when submitting articles to the journal. This document acts as an example of what this can look like and a short references for all the required and optional parts.

1.1 Document Parts

We have included a most minimal article in listing 1 that you can adapt to the concrete work to be submitted.[1] First, you have to select the programming class as in line 1. The parameter english is for language selection (optional, but recommended). The other parameter, submission, is crucial for the reviewing process and must be included for submission. More information on document parameters can be found in section 4.1. After that, wce recommend that you load the biblatex package. While it is perfectly fine to use plain BibTeX, we recommend biblatex, as it has native Unicode support and can handle URLS and DOIS better. In both cases, we have preloaded a bibliography style.

At that point, you can load any LaTeX package you need. Note that we already load some important packages with the programming class and that there are certain packages that are not recommended or incompatible with this class. Please refer to section 4.6 for information on which packages are recommended. After that, you start the actual document with \begin{document}.

You have to provide details to your submission that will be used during the review process. For that, use the paperdetails command and specify *perspective* and *area* of your submission. For perspective, choose form one of the *Art*, *theoretical* or *empirical Science*, or *Engineering* and provide it as argument to the perspective keyword; they can be found in section 2.1. For area, specify an argument to the area keyword (in braces). Find a list of suggested areas in section 2.2.

We need you to provide certain metadata which you can do with the following commands, as form line 12. You can specify the title and maybe the subtitle with \title (and \subtitle, respectively). Then please list all authors. For that, use one \author command per author and give an affiliation for each with \affiliation. When several consecutive authors share the same affiliation, you should only use one \affiliation command after the last of them. More information can be found in section 4.2.

Please include one to five keywords, separated by commas, that are important to your work with the \keywords command as in line 18. We kindly ask you to classify your work using the *ACM Computing Classification System*.[2] Please visit the ACM's website and generate the TeX code that matches your classification. The code should consist of a CCSXML environment and several \ccsdesc commands, as in lines 20 to 30.

[1]Please note that all article source must be UTF-8 encoded, no exceptions.
[2]https://dl.acm.org/ccs/ccs.cfm.

Listing 1 A very minimal article.

```
\documentclass[english,submission]{programming}
\usepackage[backend=biber]{biblatex} % Use Biblatex
\addbibresource{example.bib}

\begin{document}

\paperdetails{
  perspective=engineering,
  area={General—purpose programming}
}

\title{The importance of why and how to do work}
\author{Anna Author}
\affiliation{The Unseen University, Ankh—Morpork}
\author{Bert Betatester}
\affiliation{Carolingian Minuschool Academy, Bodoni, San Serriffe}

\keywords{paper, showcase, lorem ipsum}

\begin{CCSXML}
<ccs2012>
<concept>
<concept_id>10002944.10011122.10003459</concept_id>
<concept_desc>General and reference~Computing standards, RFCs and guidelines</concept_
    ↪ desc>
<concept_significance>300</concept_significance>
</concept>
</ccs2012>
\end{CCSXML}

\ccsdesc[300]{General and reference~Computing standards, RFCs and guidelines}

\maketitle

\begin{abstract}
  This paper shows...
\end{abstract}

\section{Introduction}
The art of computer programming~\cite{DBLP:journals/cacm/Knuth74}...
\section{...}
...
\acks
I want to thank ....

\printbibliography
\end{document}
```

With \maketitle and the abstract environment, you can produce the article's title page. Then you can start with your content. If applicable, you can put acknowledgements at the end after an \acks command, as in line 42.

This should cover the basic usage.

1.2 Running LaTeX

This class has been tested with the popular LaTeX programs available. It should work with pdfLaTeX, LuaLaTeX, and XeLaTeX. We suggest that you use LuaLaTeX, which is included in all major TeX system installations, has a good Unicode support, and is well portable. Please do not use the LaTeX/dvips combination to produce output files. To get all metadata right, several LaTeX runs will be necessary. Hence, a typical run looks like this:

```
lualatex myarticle
biber myarticle
lualatex myarticle
lualatex myarticle
```

You should replace lualatex with the LaTeX program of your choice, and biber with bibtex if you do not want to use Biblatex (or use Biblatex's BibTeX backend).

Your PDF file for your article should then be ready for submission.

2 Submission Classification

Almost anything about programming is in scope for The Art, Science, and Engineering of Programming. However, you should classify your article according to the following aspects.

2.1 Perspective

We accept descriptions of work under different perspectives:

The Art. This perspective is about knowledge and technical skills acquired through practice and personal experiences. Examples include libraries, frameworks, languages, APIs, programming models and styles, programming pearls, and essays about programming.

Science (theoretical). This perspective is about knowledge and technical skills acquired through mathematical formalisms. Examples include formal programming models and proofs.

Science (empirical). This perspective is about knowledge and technical skills acquired through experiments and systematic observations. Examples include user studies and programming-related data mining.

Engineering. This perspective is about knowledge and technical skills acquired through designing and building large systems and through calculated application of principles in building those systems. Examples include measurements of artifacts' properties, development processes and tools, and quality assurance methods.

To classify your work as any of the above perspectives, please set the perspective keyword of the \paperdetails command to one of the following values:

art for submissions concerning *The Art* (theart is a proper alias);

sciencetheoretical For submissions concerning *Science (theoretical)*, with aliases sciencetheoretical, theoreticalscience, theoretical, science-theoretical, and theoretical-science;

scienceempirical for submissions concerning *Science (empirical)*, aliases are scienceempirical, empiricalscience, empirical, science-empirical, and empirical-science; and

engineering for submissions concerning *Engineering*.

2.2 Area of Submission

Independent of the type of work, the journal accepts submissions covering several expertise areas. Expertise areas include, but are not limited to:

- General-purpose programming
- Distributed systems programming
- Parallel and multi-core programming
- Graphics and GPU programming
- Security programming
- User interface programming
- Database programming
- Visual and live programming
- Data mining and machine learning programming, and for programming
- Interpreters, virtual machines and compilers
- Modeling and modularity
- Testing and debugging
- Program verification
- Programming education
- Programming environments
- Social coding

To specify the area of expertise for your work, please set the area keyword of the \paperdetails command to one of the preceding suggested areas, or provide your own. Please use curly braces {} around the value.

3 Typographical and Technical Aspects

We have carefully chosen the presentation for articles published in The Art, Science, and Engineering of Programming to fit on-screen reading and occasional printouts. It is important to maintain a uniform look of all articles appearing in the journal. We kindly ask you therefore to *not* change the presentation. Most importantly,

- please do not change the font or the font size for that matter. The text font remains fixed as *Charter* (in the XCharter variant) and emphasis font is *Fira Sans*. The only exception is the typewriter font, where you can choose from three variants (please refer to section 4.1). Moreover;
- please do not change the margins or the line spacing;
- please refrain from using the `\sloppy` command, especially for the whole document;
- please avoid the use of the `[H]` or `[h!]` modifiers for figures;
- please use a consistent formatting (for example, use the siunitx package [6].)

This class has been tested on several LaTeX systems. We suggest that you use an up-to-date system, for example TeX Live,[3] which is available for Linux, Windows, and Apple systems, among others. We tested with the 2015 and 2016 systems and will make sure that it will work with upcoming systems, too. Please first upgrade your system if problems with this class should occur. Regarding the LaTeX system, please keep in mind to

- use LaTeX packages only if required, as fewer packages lower the chance of conflicts (refer to section 4.6 for details);
- keep the use of custom macros low to not interfere with provided packages and use LaTeX commands to do so (for example, `\newcommand` and similar instead of TeX's `\def`);
- watch out for unsuitable line- and pagebreaks;
- when using pixel graphics, provide at least 300 dpi imagery, PDF files preferred;
- minimize the use of LaTeX comments, especially via block constructs;
- consult the *l2tabu* documentation [5] about outdated packages;
- provide DOIs for all your references, if possible.

We suggest the use of a spellchecker to avoid typos.

4 Detailed Reference

Several aspects of this class as outlined above provide for some fine tuning.

4.1 Class Options

As explained in section 1, submissions to the journal must use the submission package option. For the accepted final, to-be-published version, this option should be omitted or replaced by crc for “Camera-ready Copy”. Actually, both are shortcuts for the phase keyword, which could be used, instead. Hence, the following statements in either column are equivalent:

[3] https://www.tug.org/texlive/.

Final phase

```
\documentclass[phase=final]{programming}
\documentclass[crc]{programming}
\documentclass{programming}
```

Submission phase

```
\documentclass[phase=submission]{programming}
\documentclass[submission]{programming}
```

Code Fonts The standard font for code, as selected with \texttt, \ttfamily, or a listing, is set to be the same as the sans serif font, *Fira Sans*. This matches the rest of the document style very well. However, it may be strictly necessary to have a non-proportional (or monospaced) font for the typewriter style. We therefore provide a class option code to change the typewriter font. It takes one the following values:

code=sf Sets the code font to the sans serif font (Fira Sans). This is the default. Aliases for this option are code=sans and code=sansserif.

code=tt Sets the code font to Fira Sans' monospaced equivalent, Fira Mono. Aliases for this option are code=mono and code=monospace. This is the preferred choice if monospaced typewriter font is necessary. Note that this font does not have italics.

Other Options All other options give in the \documentclass command are not processed directly by the class but rather passed on to other packages as "global options". For example, the english option from the example above does not influence the programming class directly, but rather will be picked up by Babel [2] to set up language options.

4.2 Document Metadata

The presentation of the document metadata can be fine tuned, as well. Besides the \title and \subtitle commands, there is the \titlerunning command to specify a shorter title when the normal title does not fit into the page header. Similarly, the \authorrunning command can be used to specify the presentation of the authors in the header.

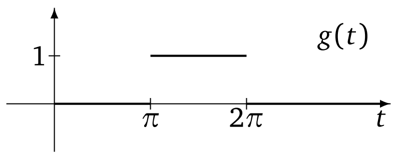
Sometime listing the affiliations of the authors can be tricky when they are shared. An optional argument to the \author and \affiliation command can be used to associate authors and affiliations explicitly. Four variants for affiliations are outlined in the beginning of listing 2.

For the final publication, we ask you to provide a short information of each author, typically a very short biography, and a photo. You can use the \authorinfo command right after an \author command to associate that author information with an author. See line 25 in listing 2 for example. The photo is given as optional argument as file name in brackets (without file extension). Omitting the photo will produce a blank space next to the author information. The actual information follows as an argument. Note that the author's *name* is prepended to the information automatically and is not to be repeated.

An author's email address can be given in either the affiliation or author information block. Please use the \email command for that purpose.

Listing 2 Four ways to specify affiliations / Author information example / Sample publication paper details

```
% Independent affiliations
\author{Anna Author}
\affiliation{The Unseen University, Ankh—Morpork}
\author{Bert Betatester}
\affiliation{Carolingian Minuschool Academy, Bodoni, San Serriffe}

% Shared affiliations
\author{Anna Author}
\author{Bert Betatester}
\affiliation{Carolingian Minuschool Academy, Bodoni, San Serriffe}

% Explicit affiliations
\author[a]{Anna Author}
\author[b]{Bert Betatester}
\author[a]{Claire Cadence}
\affiliation[a]{The Unseen University, Ankh—Morpork}
\affiliation[b]{Carolingian Minuschool Academy, Bodoni, San Serriffe}

% Mixed affiliations
\author[a]{Anna Author}
\author[a,b]{Bert Betatester}
\affiliation[a]{The Unseen University, Ankh—Morpork}
\affiliation[b]{Carolingian Minuschool Academy, Bodoni, San Serriffe}

% Author information
\author{Anna Author}
\authorinfo[anna]{is a researcher at the Unseen University in Ankh—Morpork and
a Ph\,D student to Rincewind. Contact her at
\email{anna.the.author@univ—unseen.edu.pratchett}}
\affiliation[a]{The Unseen University, Ankh—Morpork}


%%%%%%%%%% Sample publication paper details
\paperdetails{
  submitted=2017—02—18,
  published=2017—08—01,
  year=2017,
  volume=2,
  issue=1,
  articlenumber=4,
}
```



■ **Figure 1** A test figure

4.3 Paper Details

Besides specifying the perspective and area of the submission, the `\paperdetails` command is also used to provide publication information for the final, camera-ready-copy phase (class option `phase=final`). In this case, perspective and area are removed from the title page and the Digital Object Identifier (DOI), as well as the dates of submission and publications are presented instead. The information necessary for this is provided *by the publisher* after acceptance. It includes the following information:

year The year of the articles publication.

volume The volume the article appears in.

issue The issue the article appears in.

articlenumber The number of the article within the volume.

submitted The date the article was submitted to (and received by) the publisher. Please use an ISO date (YYYY-MM-DD).

publisher The date the article was published and made public by the publisher. Please use an ISO date (YYYY-MM-DD).

See the end of listing 2 for an example.

4.4 Figures, Tables, Listings

Sometimes it is necessary to pay special attention to the *floating elements* of your document, that is figures, tables, listings, and similar. While it is tempting to force those elements to the specific position where they are written down, this is not always the best choice, especially since it can interrupt the reading flow. Therefore please refrain form using the `[h!]` and `[H]` specifiers for such elements. If you *must* have the element at the current position, it maybe is not a *floating element*, in the first place. It is perfectly fine to use graphics *without* a `\begin{figrue}…\end{figure}` environment or use a `\begin{tabular}…\end{tabular}` *without* an enclosing `\begin{table}…\end{table}`.

The caption position of a floating element should match its content. For figures, graphs, graphics, or similar, please us captions *below* the imagery, as in figure 1. For tables, listings, algorithms, or similar list-like content, please use captions *above* its contents, as those elements are typically read from top to bottom. Find examples in the listings already presented and table 1.

For tabular content, we suggest to avoid vertical lines altogether and keep horizontal lines to a minimum. Refer to the `booktabs` package documentation for more information [3]. Also, the `threeparttable` package is valuable if you need footnotes in your table [1].

Table 1 Differences between things projected and things achieved

Part	done	Value	Unit
Title	yes	2.3456	dB
Abstract	yes	10^3	kHz
Rest is not entirely true			
Context	yes	90.473	%
Problem	no[a]		
Solution	yes	5642.5	MB
Implementation	yes	1.2×10^{-3}	m^2/s
Evaluation	no		
Related Work	no		
Conclusion	yes	4955.3	/kg

[a] Just a few things missing

4.5 Bibliography

As outlined, the programming class supports both plain BibTeX as well as the newer Biblatex bibliography package to present your references. Please note that you may not change the bibliography style, it is pre-selected by the class. It is a simple, numbered style with all references sorted alphabetically. In text, it should look like this: [4]. Note that the reference has the DOI included. We would like to encourage you to always provide the DOIs of the works you cite, if possible. However, please make sure in your bibliography file, that the doi entry is a plain entry without a "resolver" (for example, http://dx.doi.org/) prepended. A typical example for an entry format should look like in listing 3. Please shorten neither names nor journal articles, if possible.

The bibliography commands for your article should look like this (for Biblatex and plain BibTeX, respectively):

Biblatex

```
\documentclass{programming}
\usepackage[backend=biber]{biblatex}
\addbibresource{example.bib}
\begin{document}
...
\printbibliography
\end{document}
```

Plain Bibtex

```
\documentclass{programming}


\begin{document}
...
\bibliography{example}
\end{document}
```

4.6 LaTeX Packages Considerations

The following packages are automatically loaded by the programming class and need not to be loaded manually.

- accsupp
- amsmath
- amstext
- array
- atbegshi
- babel
- booktabs
- calc
- caption

Listing 3 A typical BibTeX entry for [4]

```
@article{DBLP:journals/cacm/Knuth74,
  Author = {Donald E. Knuth},
  Doi = {10.1145/361604.361612},
  Journal = {Communications of the {ACM}},
  Number = {12},
  Pages = {667——673},
  Title = {{Computer Programming as an Art}},
  Volume = {17},
  Year = {1974}}
```

- colortbl
- comment
- csquotes
- doclicense
- expl3
- FiraSans
- fnpct
- fontspec[4]
- fontenc
- gettitlestring
- graphics
- graphicx
- grfext
- hypcap
- hyperref
- hyperxmp
- listings
- mathdesign
- microtype
- morewrites
- multirow
- relsize
- rotating
- siunitx
- subcaption
- tabularx
- textcase
- textcomp
- threeparttable
- typearea
- url
- verbatim
- xcolor
- xparse
- xspace
- xunicode

The following packages *must not* be loaded.

- SIstyle
- SIunits
- a4wide
- a4
- aecompl
- ae
- caption2
- courier
- doublespace
- epsfig
- epsf
- euler
- fancyhdr
- fancyheadings
- fourier
- geometry
- glossary
- helvet
- isolatin
- mathpple
- mathptmx
- mathptm
- newtxmath
- newtxtext
- palatino
- psfig
- pslatex
- scrpage
- subfigure
- subfig
- t1enc
- times
- umlaut
- utopia
- zefonts

Additionally, any package that changes the font must not be loaded.

[4]LuaLaTeX / XeLaTeX only

Acknowledgements We want to thank you for using this class to prepare articles for The Art, Science, and Engineering of Programming.

References

[1] Donald Arseneau. *Tables with captions and notes all the same width*. CTAN: macros/latex/contrib/threeparttable. June 13, 2003.

[2] Javier Bezos and Johannes L. Braams. *Multilingual support for Plain TEX or LATEX*. CTAN:macros/latex/required/babel/base. Apr. 23, 2016.

[3] Danie Els and Simon Fear. *Publication quality tables in LATEX*. CTAN:macros/latex/contrib/booktabs. Apr. 27, 2016.

[4] Donald E. Knuth. "Computer Programming as an Art". In: *Commun. ACM* 17.12 (1974), pages 667–673. DOI: 10.1145/361604.361612.

[5] Mark Trettin and Marc Ensenbach. *Obsolete packages and commands*. CTAN: info/l2tabu. Feb. 3, 2016.

[6] Joseph Wright. *A comprehensive (SI) units package*. CTAN:macros/latex/contrib/siunitx. Mar. 1, 2016.

A A Famous Filler Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

About the authors

Tobias Pape is the author of this LaTeX class. Contact him at tobias.pape@hpi.uni-potsdam.de.

Cristina V. Lopes is associate editor for the first two issues of The Art, Science, and Engineering of Programming. Contact her at lopes@ics.uci.edu.

Robert Hirschfeld is chair of the AOSA steering committee. The Art, Science, and Engineering of Programming is published by AOSA. Contact Robert at hirschfeld@hpi.uni-potsdam.de.