Writing Computer Engineering at University

GUIDES FOR WRITING IN SPECIFIC DISCIPLINES

Universitat Abat Oliba CEU
UAB Universitat Autònoma de Barcelona
Universitat de Barcelona
Universitat de Girona
Universitat de Lleida

1 What is computer engineering?

No single definition of computer engineering satisfactorily encompasses all its dimensions. One might say, however, that it is the branch of engineering that applies the basic principles of the computational sciences, electronics and software engineering to develop integrated computing and communications solutions for automatic information processing.

Computer engineering students use computers and algorithms to work on concepts and ideas, software and hardware designs, and applications, and study the impact of these on society. In addition, computer engineering aims to design and implement innovative software and computer solutions in emerging areas of work.

2 General features of writing in computer engineering

Computer engineering texts should be **objective**, with writing that is neutral, clear and well **structured**, as in the following paragraph, with good sentence syntax, the proper use of connectors and precise language.

> The recent increase in popularity of virtual assistants, or chatbots, in the field of human-machine interaction has led to a proliferation of platforms that facilitate their design and integration. Virtual assistants are non-human conversational agents designed to communicate through simple, short conversations with specific objectives. They are, therefore, not useful when the aim is to establish longer, more varied and complex conversations with users.

Precision and **rigour** should always characterise computer engineering texts. For example, saying "this algorithm has an order of complexity of $O(n)$" is preferable to saying "it is an efficient algorithm", since the first expression is much more precise and rigorous.

Correct use of verb tenses. The present tense is most common in academic papers and oral presentations. Use it to state facts and analyse data. Use the past tense only to refer to completed actions in the past and the future tense only to refer to results that are expected or events that have yet to occur.

3 Typical text types in computer engineering

Labs. Practical training sessions are common features of computer engineering studies. Often, at the end of such training, you will have to present a brief report explaining what it consisted of and what resulted from it. The different steps taken and the results obtained (if any) should be included in a clear and organised way.

Final thesis. At the end of undergraduate computer engineering studies, you must submit a final written thesis. In it, you will follow a prearranged structure, with required sections, such as a title page, an index, an abstract and key words, the body of the text and a bibliography, as well as other optional sections, such as acknowledgements and the list of abbreviations used or the annexes.

The title page usually includes the title of the thesis, the first and last names of both the author and the supervisor, the academic year or the date of submission, the name of the subject and the name of the programme of study.

The body of the text consists of various obligatory sections, such as the introduction and the conclusions, and other optional sections, such as the methodology or a discussion of the results.

When writing the project report, you will find the points below to be helpful:
- Follow the natural order of sentence elements.
- Write cohesive and well-structured sentences and paragraphs, using connectors and punctuation correctly, and including only necessary and relevant information.
- Prefer affirmative sentences.
- Minimise the use of nominalisations and unnecessary possessives.
- Use precise words to avoid ambiguity.

Before submitting the final report, read the entire text one last time to ensure that all the images and tables appear in the right places, that it contains all the content and sections it should, and that it is free of spelling errors and grammatical mistakes. For this, use the spelling and grammar checking tools included in your word-processing program.

Oral presentations. These must be well organised and structured, and given in an appropriate register. In addition, thematically, they should correspond fully with the written report. The ideas must be well connected and expressed clearly. Ensure that the terminology you use is correct and you demonstrate your mastery not only of the topic but also of the grammar.

The supporting material you choose, whether it be a poster or a slide presentation, must be well written (correct use of language) and legible at a distance.

> Submit all documents in a non-editable output format, such as PDF, to avoid unintended changes in their configuration or unwanted text modifications made by the computer program used to open them.

4 Writing conventions in computer engineering

How to insert tables and illustrations. Number all the tables and illustrations in the text for easy reference. In addition, give each one a caption, usually written under the table or illustration and including the data source or other complementary information.

If you have to create a table, we recommend the following:
- Adjust the width of the columns to the size of the data they contain.
- In the cells, left justify text and right justify figures, including in the headings of corresponding columns.
- Round the figures to one or two decimals unless precision is specifically required. To indicate numbers very precisely, use scientific notation.
- Use bold to mark any text you want to highlight.

Colour	*t* value	*p* value
Orange	–2.48	0.01
Green	–3.49	0.00
Purple	–0.89	0.37

How to write the names of programming languages, operating systems and applications. Write programming languages in word-initial roman capital letters (uppercase) and the corresponding initials, if relevant, in roman capital letters.

The Python language, the hypertext markup language or HTML

Write the names of operating systems and computer applications in roman type with appropriate capitalisation in accordance with the corresponding trademark.

OpenOffice, Linux

How to embed code in the middle of text. Although we do not recommend you embed code in the middle of text, at times it may be necessary to specify some aspect of it or to highlight part of an algorithm. In these cases, it is preferable to embed the code

as text (not as an image) and to write it in Consolas font the same size as the rest of the text. In addition, highlight the syntax, use indentation and number the lines for easy reference. Each line of code should be no longer than 80 characters. In any event, the body of the text and the spacing between lines must allow for easy reading of the code. Below is an en example in Python 3.6.

```
1 squares = {}
2 for x in range(6):
3     squares[x] = x*x
4 print(squares)
```

Some programs apply this format automatically with a label whose attributes you can define as you wish. For example, in HTML the label is <code> while in LaTeX the package `listings` is used. (See the detailed explanation below.)

If you embed the code as an image, insert an alternative text with the same code to facilitate its accessibility.

To write your code correctly, consult the style guide of the programming language you work in: PEP8 and PEP257 for Python, the Google Java Style Guide for Java, the Google C++ Style Guide for C++, or others indicated by professors.

How to embed equations and formulas. Write the equations and formulas included in your document in LaTeX or insert them using the corresponding word-processing option. In either case, take into account the following aspects when defining them:

• Write the symbols of mathematical variables, physical magnitudes and mathematical functions in italics without a final point (full stop).

$x = 2b\ (4 - c)$
$F(x, y)$

• Write the symbols of mathematical constants, of the units and of the special mathematical functions (trigonometric and logarithmic) in lowercase roman letters and without a final point (full stop).

e = 2.718 281 [base of the Naperian logarithms]
tg x [tangent of x]
lb x, log2 x [logarithm to the base 2 of x]

• Write the symbols of arithmetic and relational operators (+, –, ×, =, <, etc.) in roman type. In the operations, leave a blank space between the different operands, except in the case of the forward slash (/), which you should write without spaces. If the sign stands alone before a number, write them together without spaces.

$\lambda \simeq 10$ nm
$34 + 52 = 86$
a/*b*
±20, –5

• Write subscripts and superscripts, which by themselves are symbols of physical magnitudes or of mathematical variables, in italics. Write the rest in roman type.

C_V [specific heat at constant volume]
V_m [molar volume]

• Leave a small space to separate the thousands and the decimal numbers (beginning with a decimal separator). Separate the decimals from the whole part with a point (full stop).

1 763.141 592 65 mg

• Leave a blank space between the symbol and the number that accompanies it.

5 ℃
15 ‰

• If you must refer to the formulas and equations in the text, number them preferentially with Arabic numbers between parentheses and place them on the right margin of the page.

$$x = 2b\,(4 - c) \qquad (4)$$

Units of measurement. Preferably, units of measurement should follow the international system (SI) and always be referred to in the appropriate scale. Write the corresponding symbols in lowercase roman letters unless the symbol originates from a proper name.

758 m
20 W

Write SI prefixes that are equal to or less than 10^3 with an initial lowercase letter, and those that are greater than 10^3 with an initial capital or uppercase letter.

10 km
4 nJ
20 GB (gigabyte)
30 Gb (gigabit)

Good LaTeX practices. For any written document, we recommend LaTeX because it avoids formatting and coding errors by requiring you to declare the document's logical structure. The packages in this system let you choose the desired visual appearance of the document. LaTeX then distributes the content of the document to create a nice visual design. The document generated is typically a PDF file. We suggest you check if any predefined templates or models in LaTeX in the department or faculty may be useful to you. Below we recommend some specialised packages.

• To include figures, use the `graphicx` package `\usepackage{graphicx}`. The figures can then be included with the commands

```
\begin{figure}
\includegraphics[options]{file-name}
\caption{text}\label{label}
\end{figure}
```

where the *file-name* is the name of the file that must be included, *options* indicates image options (normally the size), `\caption` and `\label` indicate the text and label, respectively, associated with the figure. In another part of the document, it is possible to refer to the figure with `\ref{label}`.

We recommend using vector formats (for example, SVG) to include figures associated with flow diagrams, class diagrams, use case diagrams, etc., and bitmap formats (for example, PNG) to include screenshots from an application.

• To include code from a program, use the listings package with `\usepackage{listings}`. Code can also be included as a figure. To include code in a document, use the commands

```
\begin{lstlisting}
[source code to be included]
\label{label}
\end{lstlisting}
```

The package supports numerous languages such as C, C++, Fortran, Python or Java. The package includes commands to indicate, among other things, the language of the source code. That allows LaTeX to format the included code according to the specified language.

• To write mathematical formulas in AMS (American Mathematical Society) format, use the `amsmath` package with the command `\usepackage{amsmath}`.

This package allows different types of formulas to be included in the text in various ways: directly within the paragraph along with the text, as an independent paragraph or as systems of equations. Users can also label the formulas with `\label` and can refer to them from any part of the document with the command `\ref.`

• For documents to be read on screen, use the `hyperref` package with the command `\usepackage{hyperref}`.

With this command, all the cross-references are hyperlinks. For example, click on a specific section in the index and the PDF viewer will automatically direct you to the appropriate page. The same is true of cross-references to figures, mathematical formulas and the bibliography.

LaTeX works on multiple platforms and is available for the Windows, Mac and Linux operating systems. Also available are on-line editors, such as Overleaf, that allow LaTeX file editing without prior installation on the computer.

Before submitting the finished document, check for errors with the spell checker set to the variety of English in which you have written the text.

In addition, when you save it, back it up in the cloud or on an external hard drive, for example.

Good practices with office automation documents. Every document created should be accessible to any person who wishes to consult it. Therefore, take care to follow these guidelines:
• Define the structure and the reading order of the document with labels and styles.
• Specify the language in which you have written the text.
• Insert any tables and numbered lists using the corresponding option, not with tab stops and blank spaces.
• Assign an alternative text to all non-text elements.
• Ensure that the combinations of background and foreground colours of any element contrast and that the size of the font used allows for easy reading.
• Check for correct spelling of all internal and external links of the document and that they work. Also, check that the size of the font used to write them is large enough so that readers can click on it easily.

If you have to convert the document into a PDF file, be careful not to lose all the introduced features. For a correct conversion, consult the Accessibility in PDFs document (*Observatorio de la Accesibilidad*) listed in section 5.

5 Selected works and websites for writing in computer engineering

1. Biblioteca terminològica, TERMCAT
https://www.termcat.cat/en/biblioteca-en-linia/biblioteca-terminologica/arees-tematiques/Informàtica.%20Tecnologies%20de%20la%20informació

Bibliographical references to dictionaries, glossaries, lexicons and dissemination materials that contain Catalan terminology with English-language equivalents from the fields of computer science and information technologies. Some resources are available online.

2. Cercaterm, TERMCAT
http://www.termcat.cat/ca/Cercaterm

Free online consultation of up-to-date Catalan terminology for most computing terms and equivalents in other languages (especially Spanish and English).

3. The Not So Short Introduction to LaTeX
https://tobi.oetiker.ch/lshort/lshort.pdf

LaTeX tutorial indicating the logical structure of a document: the title, the authors, the abstract, the sections and subsections, and the bibliography. It also explains how to include figures, tables or mathematical formulas.

4. Complementary Word dictionaries
http://www.ub.edu/sl/ca/alt/recursos/diccionari

UB Language Services page with complementary dictionaries and exclusions for MS Word and other spell checkers.

5. Accessibility in PDFs, *Observatorio de la Accesibilidad*
https://administracionelectronica.gob.es/pae_Home/pae_Estrategias/pae_Accesibilidad/pae_documentacion/pae_eInclusion_Accesibilidad_de_PDF.html?idioma=en#.XabYHUkUmUk

Recommendations for creating accessible documents.

6. Treballs acadèmics
https://www.ub.edu/cub/treballs

The section of the Criteris de la Universitat de Barcelona (CUB) covering the bachelor's degree final project.

7. Writing in Electrical and Computer Engineering: A Sampling of Advice from Faculty, Marquette University
https://www.marquette.edu/wac/departmental/MarquetteUniversityWritinginElectricalandComputerEngineering.shtml

This site includes general advice about types of assignments, features of good writing, valid evidence and citation formats, as well as specific advice about electrical and computer engineering texts. In addition, links at the bottom of the page lead to information about paraphrasing to avoid plagiarism.

8. How to write a computer science paper, Lorenz Froihofer
https://www.froihofer.net/en/students.html

The menu on the left-hand side of the "For students" page of this website provides general advice on thesis and scientific writing, and more specific advice on writing a computer science paper and on using LaTeX.

9. Style Guide: Electrical and Computer Engineering, University of Nebraska-Lincoln
https://engineering.unl.edu/downloads/files/ECE-Style-Guide.pdf

Although specifically intended for Electrical and Computer Engineering students at the University of Nebraska-Lincoln, this guide provides potentially useful advice on writing well and formatting texts properly.

10. Technical writing, Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand
https://www.cosc.canterbury.ac.nz/teaching/4thyear/technical_writing/

Hints and tips for more successful technical writing. Use the menu on the right-hand side of the page for useful advice on style and grammar.

11. Helping engineers communicate/Use the CommKit, Communication Lab, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
http://mitcommlab.mit.edu/eecs/use-the-commkit/

"The CommKit is a collection of guides to successful scientific communication, written by the Communication Fellows of MIT's Department of Electrical Engineering and Computer Science." The menu on the right-hand side of the page leads to technical communication and visual communication advice.

Servei de Llengües (UAB), Servei de Llengües Modernes (UdG), Serveis Lingüístics (UB), Servei de Llengües (UAO CEU) i Institut de Llengües (UdL)

Coordinació: Virgínia Castillo i Enric Serra
Autora: Laura Llahí (amb el suport de Rosa Gual i Marta Juncadella)
Assessorament: Lluís Garrido i Mireia Ribera
Agraïment a: François Malabre
Revisió lingüística: Pere Drou, Andreu Pulido
Adaptació a l'anglès i revisió lingüística: Peter Redmond
Aquest projecte ha rebut un ajut Interlingua de la Generalitat de Catalunya
Data d'edició: octubre de 2019