Instructions for using the thesis template

This document is a template on which you can write your final thesis at LUT University. It meets the requirements for the content, structure and layout of the BSc and MSc theses as well as the accessibility criteria for material published in the LUTPub repository. It is especially important that you use the template in formulating the title page, abstracts, table of contents, list of references and citations.

The template is applied in all BSc and MSc thesis projects that start after 11 August 2021.

The actual thesis template starts on the following page. Write your own thesis using this template and delete the introductory texts (e.g. this help page) and other content in the template that you do not need.

The attachments include important instructions – remember to read them as well. The thesis supervisor can give additional instructions.

In addition, read other thesis guidelines released by the Lappeenranta–Lahti University of Technology LUT:

• Thesis instructions in the UNI portal

• LUT Academic library, pages for thesis students and researchers

• LUT Academic Library, information retrieval guide

• LUT Academic Library, copyright guidelines

• LUT University Code of Conduct

• Plagiarism detector Turnitin guidelines for LUT students

• Graduation process and instructions in the UNI portal

UTILIZING CODEGRADE TO BUILD AN INTRODUCTORY WEB PROGRAMMING COURSE

Lappeenrannan–Lahden teknillinen yliopisto LUT

Bachelor's thesis, Software Engineering

2022

Author: Vili Huusko

Supervisor: University Lecturer Erno Vanhala (D.Sc.)

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Engineering Science

Software Engineering

Vili Huusko

Utilizing CodeGrade to build an introductory web programming course

Bachelor's thesis

Year of completion of the thesis

xx pages, xx figures, xx tables and xx appendices

Examiner(s): University Lecturer Erno Vanhala (D.Sc.)

Keywords: List keywords that define the content of the thesis and help find it.

Use single line spacing in your abstract and justify the text as you would the rest of your thesis text body. The abstract and its identifying information must fit on one A4 page. The abstract is a public document, and therefore, you need to exclude any confidential information from it.

The abstract is an independent summary of the thesis and should be intelligible as such, without the original document.

A good abstract is written in complete and concise sentences. Do not express personal opinions in the abstract – describe the thesis as an outside reporter would. Do not make detailed references to the original text. Present the key results of the research.

If you have not received your basic education in Finnish or Swedish and are writing your final thesis in English, write your abstract only in English.

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tietotekniikka

Vili Huusko

Web-ohjelmointikurssin rakentaminen CodeGradea hyödyntäen

Tietotekniikan kandidaatintyö

xx sivua, xx kuvaa, xx taulukkoa ja xx liitettä

Tarkastaja(t): Yliopisto-opettaja Erno Vanhala (Tkt)

Avainsanat: Listaa tähän avainsanat, joiden perusteella tutkielma voi haettaessa löytyä

ACKNOWLEDGEMENTS

(Optional)

You may thank people who have supported you in your work on the thesis.

Abbreviations

AAT	Automated Assessment tool
API	Application Programmable Interface
CDN	Content delivery network
CLI	Command line interface
CMS	Content management system
CSS	Cascading style sheets
HTML	Hypertext markup language
I/O	Input/Output
JS	JavaScript
JSON	JavaScript Object Notation
REST	Representational state transfer

Table of contents

Figures

Tables

NOTE! Lists of figures and tables are optional.

1. Introduction

The introduction explains the reasons for the research and what it aims to achieve. It also gives a brief overview of the methodology used and key results. The first paragraph should answer the following questions: What is your thesis about? Why is this topic important to study right now? Why is it important in general? The first paragraph entices people to read your thesis.

The second paragraph briefly presents the key findings of previous studies, and the third one explains what previous research has not taken into consideration or examined. In other words, it explains why your study is necessary. Conclude your introduction by relating the objectives, research problem, research questions, limitations and thesis structure briefly.

Lappeenranta-Lahti university of technology (LUT) as summer of 2022 offers only one mandatory web-programming course for the computer science department called *Web applications* which gives the students the mandatory tools and knowledge to build full-stack applications. The course is, however, set to be divided into separate courses for the 2022-2023 term. The course will be divided into *Introduction to web programming* and *Advanced web programming* based on the level of difficulty of the topics covered. The upcoming *Advanced web programming* is going to inherit most of the content from *Web applications* but the content to the new *Introduction to web programming* has to be made from ground-up. New exercises have to be built for the course and the assessment of the exercises have to be made utilizing the CodeGrade automated assessment tool (AAT).

The *Introduction to web programming* course will cover the basics of client-side (front-end) web development. It will teach the students how to write Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and several JavaScript frameworks, such as Frappe-chart.js and Leaflet. The course also teaches how to make application programmable interface (API) calls and how to fetch and use JavaScript object notation (JSON) data.

The *Advanced web programming* course will focus on full-stack development. In addition to front-end development, it also consists of back-end (server-side) programming. Students

will get familiarized with concepts such as Node.js JavaScript run environment and React.js front-end framework.

The intensiveness of the previous *Web applications* course was a factor in splitting the course into two parts to smooth out the difficulty curve. The course saw a major drop in attending students in week 4 during the introduction of Node.js back-end programming. The course feedback suggested that many students found the exercises of week 4 overwhelming and the increase in difficulty too steep.

1.1 Objectives and goals

This thesis focuses on the making of the exercises to the new *Introduction to web development* course and utilizing CodeGrade AAT and Cypress.io testing framework in the assessment of the exercises. Improving the original exercises from the *Web applications* course is important to achieve a more gradual increase in difficulty. The exercise also cannot be too easy in order to be engaging and promote learning.

As mentioned previously, the *Web applications* course will be separated into 2 different courses. While the *Advanced web programming* course will need some new exercises, this thesis will only focus on the exercises and automatic assessment of the *Introduction to web development* course.

The main research questions for this thesis are

- How to construct new exercises for the upcoming web programming course?
    - How to make the difficulty curve less steep?

And

- Is CodeGrade and Cypress.io suitable for assessing the programming exercises?
    - How to implement CodeGrade with Cypress.io to automatically assess the programming exercises?

The research method used in this thesis is design-science research method. Design-science research methodology (DSRM) is used when aiming to solve a problem through creating an artifact [1]. In this case the problem is the too steep learning curve of the previous *Web*

applications course, and the artifact is the programming assignments, including instructions, prototypes, and automatic assessments of the assignments.

1.2 Structure of the thesis

This thesis consists of five chapters. The first chapter is the introductory part of the thesis. It gives a small summary about the topic of the thesis and why the research was done. It also holds the thesis' research questions and remarks on how the research was done and what method it was done with. The second chapter focuses on AATs in general, previous research regarding them and overview of the technologies used and why the technologies were chosen.

The third chapter covers the upcoming *Introduction to web development* course and its structure and what is expected from a course like that. It also covers the programming exercises and the used technologies in each week's exercises. Chapter four will focus on the testing itself. It discusses what CodeGrade and Cypress.io have to offer, how the tests are made, what is being tested and how the tests integrate to Moodle and if they have some limitations.

The fifth chapter discusses key findings during the research, and the created artifact – the exercises for the course and the tests for the exercises. The thesis concludes by summarizing the created artifact, key findings and discussing what the future holds for the research.

2. Automated assessment of programming exercises

Give a brief introduction to what each chapter is about. Write at least two sentences under the chapter heading.

This chapter covers automated assessment and its applications in education environments. Furthermore, the chapter introduces different automated assignment systems and their relevant features regarding the testing of web applications and if they're suitable for a website programming course. Additionally, the basics of CodeGrade is covered as well as why the technology was chosen over several other competitors.

2.1 First section

I skipped straight to third section, but I left the placeholder text.

Give each chapter and section a title that describes their content. You need to write something under the heading. In other words, you cannot have one heading after another without text in between. This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section.

> "This is an example of a direct citation where the line spacing can be smaller than in the rest of the text body. This is an example of a direct quote. This is an example of a direct quote. This is an example of a direct quote. This is an example of a direct quote. This is an example of a direct quote." (Source 2021, 45.)

This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section. This is an example sentence for the first section.

2.1.1 First subsection

This is an example sentence for the first subsection. This is an example sentence for the first subsection. This is an example sentence for the first subsection. This is an example sentence for the first subsection.

2.1.2 Second subsection

Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

2.1 Second section

Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text

3. Introduction to web programming

The main reason for why the *Web applications* course was divided into two different courses was the intensity of the course and the steep learning curve. The weekly exercises of *Web applications* consisted of 12 weeks, of which only 3 covered the basic topics – the same topics the whole *Introduction to web programming* will cover. In the fourth week the focus shifted to back-end programming, which introduced a whole new dimension to web programming.

This intense start was a reason many students ceased to submit anything during later weeks. During weeks 1-2 on average 80% of enrolled students submitted the weekly exercise. On week 3 and onwards an average of 50% enrolled students submitted the weekly exercise [2].

The structure of the upcoming course is important in order to make the difficulty level more gradual. In addition, the assignments need to have a precise level of challenge in order to be enjoyable but not too overwhelming. The assignments must also be able to be tested using the AAT and relevant technologies used in this course.

3.1 Requirements for the course

As *Introduction to web programming* is an introductory course for web development, it focuses only on the client sided front-end that is run on the browser. Because of this reason front-end frameworks, such as React, Angular or Vue aren’t covered. Instead, only the essential languages: HTML, CSS and JS are covered.

Nowadays, almost every website communicates with some server, usually in the form of Hypertext Transfer Protocol (HTTP) GET or POST requests. For this reason, HTTP GET, and POST requests are covered thoroughly. HTTP requests are usually sent to a Representational state transfer (REST) server, that works as an Application Programmable Interface (API). The communication with these types of servers is done utilizing JSON data format. For that reason, JSON is in a crucial role in the course’s repertoire.

JavaScript has a multitude of different libraries that can be used easily on the front-end. They are easy to include in the application with the HTML script tag with great performance when fetched from a geographically distributed network of content providing servers called Content Delivery Network (CDN) [3]. The course aims to get students familiarized with a few different frameworks to see what JavaScript has to offer.

The course assignments need to also be testable using an AAT and follow the limitations of the used technologies. The tests must always work and have the same expected output. Therefore, the assignments cannot have any elements of randomness in them. Also, the assignments can't have any visual testing, such as animations or certain pictures in them, since Cypress.io isn't able to test them. Lastly, certain CSS selectors, like hover, cannot be tested because Cypress only simulates the cursor.

3.2 Course structure

The structure of *Introduction to web development* is carefully thought out in order to avoid the previous pitfall of having too many topics in too little time. The structure of the course is progressive, where new knowledge is built upon old knowledge. With this approach, the fundamentals are rehearsed multiple times throughout the whole course.

The course also starts new topics with a quiz-type task which tests how well the student has studied the topic. The programming tasks start with smaller and easier exercises that are faster to complete, and gradually rise the difficulty in order to test how well the student can apply their knowledge into practice.

3.2.1 HTML and JavaScript

During the first two weeks, the course has its focus on HTML and JavaScript. They are the building blocks of all websites. HTML is used to create the basic structure of the website, whereas JavaScript is used to make it interactive and interact with other websites or servers.

The assignments during the first week are simple enough to make the subjects easily approachable. The assignment consists of simple tasks, such as creating an HTML element,

printing text to console, making Document object model (DOM) changes (changes to the website structure) and creating new elements. These tasks can be accomplished by using only few lines of code and little effort, in order to have the focus of the task on the languages themselves.

The second week assignments assume that the students are already familial with the HTML + JS environment. Therefore, the tasks are more advanced than in the first week. The second week has assignments that require the students to use some of the more commonly used HTML elements, such as tables, and more advanced JavaScript methods as well as images opposed to only text.

3.2.2 CSS and requests

Later in the course the students will also be implementing styling to their websites with CSS. Additionally, the tasks require to make HTTP requests to different REST APIs, mainly to Statistics Finland (Statfin) API. In week 3, the students are required to make a table, that holds data fetched from Statistics Finland. The table is also styled with CSS using different selectors to achieve a pleasant look.

Week 4 introduces web fonts, responsive design, and Uniform Resource Locator (URL) parameters. The week's task is to create a TV show viewer website, that has a search function. After the user searches for shows, the website connects to the TvMaze API using the search word as a URL query parameter.

The website will also use responsive design, which means that the website adjusts itself depending on the size of the screen it is viewed with. This is done using CSS media queries. Web fonts are fonts that are not preinstalled to the end-user's browser [4]. An example of such font is Roboto. The fonts need to be downloaded to the user from a server, usually a CDN.

3.2.3 Frameworks

The last two weeks focus on different frameworks and HTTP requests. Week 5 is built around the chart-drawing framework called *frappe-chart.js*. Additionally, the data shown in the chart is fetched using HTTP POST requests. The first two tasks focus on sending a static POST body to the Statfin API. On the subsequent tasks the POST body is modified by the user input to fetch different data from the site. The week also covers how navigation is implemented in HTML + JS.

The sixth week focuses on Leaflet interactive map framework, as well as GeoJSON, a specially formatted JSON used to display geographical data.

> "GeoJSON is a format for encoding a variety of geographic data structures using JavaScript Object Notation (JSON) [RFC7159]. A GeoJSON object may represent a region of space (a Geometry), a spatially bounded entity (a Feature), or a list of Features (a FeatureCollection)." [5]

The assignment consists of fetching geographical data of the municipalities of Finland and coloring them according to their net migration. The map also will have some interactivity to it: if the user hovers his cursor over a municipality polygon, the name of the municipality will show up. If the user clicks on one of the municipality polygons, the net migration and emigration statistics will show up. The assignment also features a JavaScript and CSS framework called Bootstrap to improve the look of the site.

3.2.4 Untestable assignments

In addition to the normal coding exercises that are tested with an AAT, the course will also include some non-testable, manually assessed assignments. On week 7 the students will learn about the canvas element and a game-making JavaScript, such as Phaser, in a form of a game-making weekly exercise.

The course will also have a section about Content Management Systems (CMS) in the form of a bonus assignment. The CMS assignment is also an assignment which isn't suitable for testing with an AAT.

4. CodeGrade and Cypress

The tasks of a traditional programming course that utilizes Command line interfaces (CLIs) are, in most cases, trivial to test. Almost every test can be conducted as an input/output (I/O) testing where the AAT only inputs some value(s) and expects a pre-determined output. In contrast, web programming needs a much wider spectrum of features from the AAT.

The website can have several elements with different properties and attributes, and these properties and attributes and even elements can change during the lifecycle of the web application. These elements also have different styling determined by CSS and these styles can change during the lifecycle of the application. The website can also fetch data from certain servers that respond data depending on the request. The request can also have a predefined body without which the server can’t respond.

To test all these features an end-to-end test must be conducted. This is easiest done using an end-to-end testing framework. CYPRESS.IO + CITATION

Something about GC

This chapter focuses on the used technologies and what they offer to achieve a good user experience for end-to-end testing.

6. Conclusions

The conclusions explain how well your research achieved its objectives, what its findings were and what they mean in a wider perspective and for the future. The conclusions should examine how your findings differ from or coincide with those of previous studies. Analyse the impact of your research: its theoretical or practical contribution and wider societal importance. In addition, mention possible limitations of your study and research topics that should be dealt with in the future.

Remember that if or when someone other than your supervisor reads your thesis, they will most likely read the introduction and conclusions first.

References

[1] J. R. Venable, J. Pries-Heje, and R. L. Baskerville, “Choosing a Design Science Research Methodology,” *ACIS 2017 Proceedings*, Jan. 2017, [Online]. Available: https://aisel.aisnet.org/acis2017/112

[2] A. Järventausta, “Piloting a new automated assessment tool in the Web Applications course,” 2022. Accessed: Jul. 03, 2022. [Online]. Available: https://lutpub.lut.fi/handle/10024/163918

[3] J. D. Gagliardi and T. S. Munger, “Content delivery network,” US7962580B2, Jun. 14, 2011 Accessed: Jul. 04, 2022. [Online]. Available: https://patents.google.com/patent/US7962580B2/en

[4] M. Nygren, *Web Font Optimization for Mobile Internet Users : A performance study of resource prioritization approaches for optimizing custom fonts on the web*. 2019. Accessed: Jul. 04, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-85481

[5] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and S. Hagen, “The GeoJSON Format,” Internet Engineering Task Force, Request for Comments RFC 7946, Aug. 2016. doi: 10.17487/RFC7946.

Appendix 1: Text processing and layout in a thesis

Good text processing skills make writing your final thesis and using this template easier. Therefore, you should make sure you have the sufficient basic skills to edit long documents with text processing software before you start. This involves applying the styles, understanding automatic referencing and knowing how to divide your text into sections.

The essential thing is to understand the following basics of Word:

- Do not modify your layout by adding consecutive spaces or line breaks. If you need to press Enter or the space bar more than once, you are probably doing something wrong. When you want to start a new paragraph, press Enter once at the end of the sentence and use styles to create a space between the paragraphs.
- Do not do any numbering manually (section numbers, page numbering, numbering of figures/tables/charts/appendices). Word has efficient automatised tools for all of this. They keep the numbers in the right order even if you modify, add or remove information.
- Do not add hyphenation at the end of a line manually. Word's automatic hyphenation tool can be used in this template. If you need to add more hyphens, select manual hyphenation in Word. The automatic hyphenation is usually turned off, but you can activate it yourself.

Line spacing, font, margins, alignment, page numbering and headings

Official layout guidelines state that the line spacing should be 1.5 except for the abstract and possible direct citations, where the spacing is 1. You can choose from two fonts: Times New Roman (12 pt) or Arial (11 pt). This template has been written in Times New Roman 12 pt. Leave an empty line between paragraphs. Also leave an empty line after tables and figures.

Leave the following margins:

- top and left 35 mm
- bottom and right 20 mm.

The page number of the title page is 1, but the page numbering should not be visible before the first page of the table of contents. Place the page numbers at the top of the page, either centred or in the right-hand corner. Page numbering ends on the final page of the reference list: appendices do not have page numbers unless the appendix is multiple pages.

Always use heading styles in your headings (Heading 1, Heading 2, Heading 3). Always place chapter headings (Heading 1) on a fresh page. If you add large figures or tables, remember to check that the empty space after the figure or image covers no more than 20% of the page.

The heading numbering starts from the introduction and continues consecutively. Do not place a period after the heading number. Do not number the heading of the reference list at the end of the thesis.

The thesis should include no more than three heading levels, and the headings should progress in a logical order (first Heading 2, then Heading 3, etc.). If you need even more detailed subheadings, do not number them, and leave them out of the table of contents. Concise headings that describe the text sufficiently are the best. You can use question marks or exclamation marks, but do not add a period if the heading is a regular sentence.

A heading cannot be followed by a heading. Always write something between them. For instance, there must be text between headings 1 and 1.1.

Lists are a good way to express things clearly. Use the same type of bullet or symbol in lists throughout your thesis. A section should never end in a list. There should always be two or three sentences after a list.

Appendix 2: References

The text must include references to the sources you use. LUT University applies the Harvard referencing style, also called the author-date style with in-text referencing and a detailed reference list at the end.

The purpose of a reference is to provide sufficient information on a source used in the study, allowing the reader to consult the original source for further information. The reference enables the reader to find detailed information on the source easily in the list of references. You should refer to the original and most recent sources. If no new studies have been published on the topic in question, also older ones may be used.

Referring to a source means that you explain the contents of the source material in your own words. Direct citations, on the other hand, are placed in parentheses (“ ”). Plagiarism or using another person’s original material without appropriate referencing is not allowed.

Referencing technique

In the Harvard system, the citation is placed in parentheses directly in the text to indicate the passage that has been cited from another source. Place the citation before the period that ends the sentence when it refers only to the sentence in question (Kaasinen et al. 2020, pp. 173–174). If you are referring to more than that one sentence, place the citation after the paragraph like a sentence of its own. In such cases, add a period at the end of the citation. (Kaasinen et al. 2020, p. 174.)

Typically, the citation mentions the author (the last name is sufficient, unless authors of several sources have the same last name), the publication year and the page number. Please note that the author does not always have to be a person, but it may also be an organisation,

for instance. If the source does not mention who the author is, the reference should include the name of the publication instead of the author. (Nykänen 2002, 77.) The author (or the title of the work) is very commonly mentioned as a part of a sentence: “According to a study conducted by Möttönen (2007, 68), a pike is a fish”.

If the source has more than one author, they are all mentioned in the reference by their last name and separated with the word ‘and’ or the symbol ‘&’. Make later references to the same work with the first author’s last name and “et al.” If you reference several works published by the same author in the same year, add lower-case letters (a, b, c…) after the publication year to distinguish the sources. Use the same alphabetical organisation also in the list of references.

Examples and detailed instructions on referencing:

LUT Academic Library’s instructions on how to cite electronic documents

Aalto University citation guide

Harvard referencing, University of Sheffield

Appendix 3: Tables, figures, equations, numbers, symbols and abbreviations

It is a good idea to illustrate your text with figures and tables. Figures and tables must have captions and consecutive numbering. The captions of tables are placed above the table and those of figures below the figure. Refer to the figures and tables in the text body, preferably before you introduce them, and align them with the text body.

Remember to add alt text (alternative text) to your figures and tables to ensure accessibility. Alt text is read with a designated reader and can be viewed even when the image cannot be displayed on the page. The MS Word text processing software creates alt text automatically, but you should make sure it describes the object sufficiently and understandably. You can modify the alt text by right-clicking on the figure or table.

Tables

Give your tables numbers and captions. Place the caption above the table, name its columns and mention the units applied, as in Table 1 below. Avoid empty columns or rows. The recommended font size is 10.

Table 1: Sensor measurements

Voltage U [V]	Pressure p [Pa]
0,984	0
2,252	150
2,772	300
3,181	450
3,615	600
3,817	750
4,088	900

Figures, charts, graphic elements

Images help illustrate your text. Number your figures and place a caption underneath. You should use a software programme such as Excel or Matlab to draw charts. Charts should be clear and easy to understand. Use a white background. A background grid is allowed if it does not make the figure difficult to interpret. Variables and measurement points should be clearly visible, as in Figure 1. Name the axes and their units.

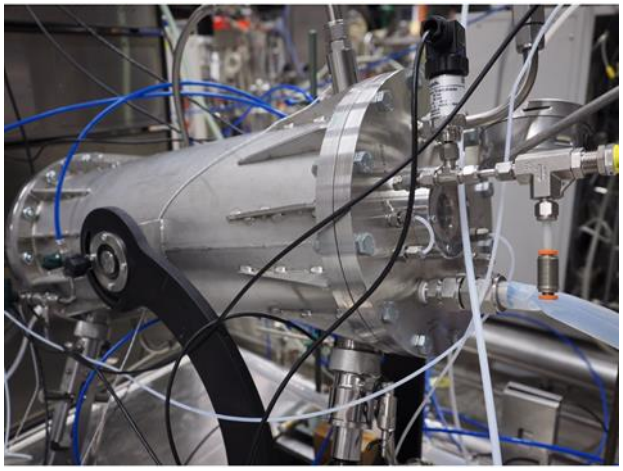

Figure 1. Gas fermentor (VTT, LUT image bank).

Create as much of the figures yourself as you can. Use the same font as in the text body and equations. If you use images created by someone else, remember to cite them correctly. Captions need to be in the same language as the text body.

Do not end a paragraph in a figure or table. Add text underneath, such as comments on the figure. Large figures, tables, long equations and other supporting material can be appended, if needed.

Numbers, symbols and equations

Numbers in the text are usually approximations. Their accuracy depends on the observational error. Include only significant figures in the results. Interim results should include at least two figures more to avoid round-off errors. Present large and small figures in powers of ten 10^n, where n should preferably be divisible by three.

Equations and other mathematical expressions must consist of standardised symbols if ones exist. You may use your own symbols only there are no applicable standardised or established ones.

Explain the symbols in an equation when you use them for the first time. Write each equation clearly on its own line and indent it. Number your equations consecutively or by paragraphs so that the number is in parentheses on the right side of the equation and aligned to the right. You can refer to an equation only after you have presented it, with certain exceptions, such as if the object you are referring to is far ahead. Example:

$$pv = RT \qquad (1)$$

where p is pressure [Pa], v is specific volume [m3/kg], R is the gas constant [J/kgK] and T is temperature [K].

When you write symbols in your thesis, do the following:

- Write all variables in italics.
- Write subscripts upright unless there is a need to italicise them. Write abbreviated subscripts and numerals e.g. as follows: $\Delta\sigma_w$, σ_1, $\sigma_{\min}$. For

instance, in the summation $\sum_i^\infty x_i$ the subscript needs to be italicised because it represents a running number $_i$.

- If you wish to express change in e.g. pressure Δp, write Δ in a regular font. In some cases, $\mathit{\Delta}$ may also be a variable and should then be italicised. π is the ratio of a circle's circumference to its diameter. $\mathit{\pi}$ may be the pressure ratio.
- Do not italicise mathematical operators such as sin *x* or lg *y*.
- Distinguish absolute values as follows: "variable_=_number_unit", with the exception of a percentage sign after a numeral, e.g. a = 5.2 mm, γ = 97.7%.

Use a decimal point (".") in accordance with international standards. In contrast, a decimal comma is used in theses written in Finnish. This also applies to figures and tables.

List of symbols and abbreviations

List symbols and abbreviations and their definitions that are not common knowledge separately on their own page before the table of contents. Divide them into groups: Roman symbols, Greek symbols, and finally, abbreviations. Give the page the heading Symbols if there are no abbreviations or Abbreviations if there are no symbols.

When you use a symbol or abbreviation in the text body for the first time, introduce it to the reader for example as follows: "The concept design for manufacturing and assembly (DFMA) is...". After this, you can use only the abbreviation, and the reader can verify its meaning from the abbreviation list. Do not add concepts to the list of symbols and abbreviations that you do not mention in your text body.

Appendix 4. Appendices to the thesis

Appendices may include e.g. interview questions, survey forms or other content relevant to the work but not necessary to include in the text body.

In your text body, refer to the appendices by adding their title in parenthesis (Appendix 1) where relevant. Give all appendices a title based on their content and list them in the table of contents in the order in which they are referred to in the thesis.

Single-page appendices do not require page numbering. Multiple-page appendices do.

Appendix 5. Publishing the thesis

LUT’s degree regulations state that Bachelor's and Master's theses are public documents. They are published in the LUTPub repository, and related instructions are available on the library web site: https://libguides.lut.fi/lutpub/opinnaytetyot.

Together with the first examiner, make sure that the commissioner of your thesis is aware of the publicity requirements from the very beginning of the discussions. If it is necessary to include information in your Master’s thesis that the commissioner wants to keep secret, the university may allow keeping the Master’s thesis confidential for up to two years. In such cases, the commissioner needs to provide the university with a notification on the confidentiality requirements. The student is responsible for submitting the confidentiality notification to Student Services no later than in connection with the assessment application. The notification must relate the scope of the confidentiality, grounds for it and the confidentiality period in full years. The confidentiality period starts from the date the thesis is assessed. Information that needs to remain confidential for over two years must be excluded from the thesis text. The thesis will be evaluated based on the version submitted to the university.

All theses in LUTPub must fulfil accessibility requirements. Your text must be as legible as possible to readers. Remember

- to use styles to create headings
- add/check the alt text of your figures and tables.
- embed hyperlinks into your text or a description of the linked content; do not include hyperlinks.