# STUDENT HOW TO

*This guide is meant to help students who plan to carry out their projects and theses at the **Institute for Visual & Analytic Computing**. We first give general guidelines for projects and theses and then briefly characterize each individual type of project or thesis in more detail.*

*The recommendations given here originate from previous experience in mentoring students. As such, these recommendations are neither set in stone nor binding in any form. For the formal and legally binding rules of projects and theses the reader is referred to the study and examination regulations in force.*

## **WHY** DO I DO **WHAT**, AND **HOW** DO I DO IT WITH **WHAT** RESULTS?

CONTENTS



*Document History*

*2021/06/10 Add section on literature for scientific writing (Christian Tominski)*

*2020/12/02 Improve formatting, re-introduce section numbering (Christian Tominski)*

*2020/11/17 Revised, remove theses, updated links, change style (Christian Tominski)*

*2015/03/13 Revised according to new regulations, updated links (Christian Tominski)*

*2012/02/16 Added Table of Contents (Christian Tominski)*

*2011/12/15 Corrected typos (Christian Tominski)*

*2011/11/23 Revised according to comments from colleagues (Christian Tominski)*

*2011/11/09 Initial version (Christian Tominski)*

# 1 GENERAL GUIDELINES

The biggest hurdle that students have to overcome when conducting projects and theses is to get familiar with the style of *scientific* working. This includes solving problems *scientifically* (i.e., following the *scientific* method), carrying out *scientific* literature reviews, and writing *scientific* reports and theses.

Although working *scientifically* may be new for you, you should not be scared. Generations of students have mastered this hurdle before. There are plenty of books, guidelines, FAQs, and other sources of information on this topic. The university offers corresponding lectures as part of the basic curriculum as well. The good student will explore and utilize these sources in preparation of a project or thesis. Here, we can only briefly summarize some of the guidelines.

## 1.1 LITERATURE REVIEW

A literature review is necessary to develop a good understanding of the state of the art with regard to a specific *scientific* problem. Only then is it possible to identify white spots in the literature yet to be filled with new *scientific* solutions. Without a literature review you risk reinventing an existing solution, which would render your own work invalid.

The literature review is an iterative process during which you collect relevant *scientific* works. The process involves phases of exploration, evaluation, and management.

### EXPLORATION

Your review usually starts with a few initial articles or books provided by your mentor. From there you have to independently explore the literature further by:

- following the references in the initial manuscripts,
- consulting the library for books on the topic,
- searching in journals and proceedings of major conferences in the field,
- conducting a web search using appropriate keywords, or
- contacting authors of articles.

### EVALUATION

The evaluation starts with identifying what a publication is about: What is the *scientific* problem addressed? What are the results achieved? How have the results been obtained? In order to get a quick glance of a work without necessarily reading all the details it makes sense to read the introduction and the conclusion first. Only if these parts are deemed to be in the scope of your project or thesis has the entire paper to be read.

The evaluation also includes separating significant work from less-relevant work. In this regard, the citation count is a good indicator, because important work is usually cited more often than insignificant work. However, care must be taken, because the citation count method does not work for very recent significant publications, which might not yet have accumulated a high citation count. In general, *scientific* works can be ranked as follows: Books go first, then come journal articles and book chapters followed by papers in conference and workshop proceedings. Last but not least, technical reports, posters, abstracts and other non-archival publications might be relevant. Web resources (e.g., blog posts, wiki pages) should be used only as a last resort if no other literature is available.

#### MANAGEMENT

All relevant publications should be managed in a database of references. There are different solutions available including online and offline variants. The format most commonly used for computer science publications is BibTeX, a format that can be exported and imported by many different tools. Moreover, it is a good basis for citations in LaTeX.

## 1.2 CONTENT OF SCIENTIFIC REPORTS AND THESES

A *scientific* report or thesis should contain the information listed below. Correct spelling and grammar as well as appropriate formatting go without saying. For all written reports it is strongly advised to use professional typesetting via LaTeX, rather than software à la Microsoft Office or LibreOffice.

- **Front matter:** title, author, reviewers and mentors, date of submission, abstract (in German and English), CR-classification, keywords, table of content, further lists such as list of figures or list of tables
- **Introduction:** brief introduction to the general topic; brief description of the addressed *scientific* problem; brief overview of the proposed solution; brief outline of the manuscript
- **Fundamentals and State of the Art:** introduction to the topic background, fundamental terms, and basic definitions; review of related work; assessment of the state of the art; identification of so far unsolved *scientific* questions
- **Concept:** main matter of the work, comprehensive *scientific* derivation and description of the solution on a conceptual level, argumentation for the proposed concept and discussion of alternatives
- **Implementation:** outline of the architecture; discussion of implementation design choices and actual design decisions, description of the implementation details
- **Evaluation:** assessment and comparative analysis of the results achieved
- **Conclusion and Future Work:** concluding and summarizing discussion of the work, identification of problems that remained unsolved, proposals for future work
- **References:** all resources used and only these must be cited properly throughout the manuscript, list all references using an appropriate format
- **Statement of Authorship:** declare authorship and sign the work

## 1.3 WRITING STRATEGY

Many students feel very uncomfortable with writing a big report or thesis of 45 or more pages. As computer science students know, the trick is to divide & conquer. First develop a general outline of the work with sections and subsections and assign to each the number of pages you plan to write. Second, take notes about the content to be written in each section. Then further break down the sections into paragraphs and describe in a single sentence what the reader is about to learn from each paragraph. This gives you a general plan and it is now much easier to formulate the content incrementally section by section.

## 1.4 LITERATURE ON SCIENTIFIC WRITING

Before engaging in a writing project, it is highly recommended to study some literature on scientific writing for two reasons. First, you will learn what is expected from you when you submit your written work. Second, by studying the literature and adhering to the given advice you will develop your own skills for good scientific writing. The following references can be used as entry points.

German language:

- H. Esselborn-Krumbiegel (2017)
  „Richtig wissenschaftlich schreiben: Wissenschaftssprache in Regeln und Übungen“
  Verlag Ferdinand Schöningh, 5. Auflage
  https://elibrary.utb.de/doi/book/10.36198/9783838547329
  (available when accessing from the university network)
- N. Franck (2019)
  „Handbuch Wissenschaftliches Schreiben: Eine Anleitung von A-Z“
  Verlag Ferdinand Schöningh
  https://elibrary.utb.de/doi/book/10.36198/9783838551081
  (available when accessing from the university network)

English language:

- T. Skern (2019)
  “Writing Scientific English: A Workbook”
  facultas, 3rd edition
  https://elibrary.utb.de/doi/book/10.36198/9783838550664
  (available when accessing from the university network)
- J. Zobel (2014)
  “Writing for Computer Science”
  Springer, 3rd edition
  https://doi.org/10.1007/978-1-4471-6639-9
  (earlier print version available in university library)

## 1.5 MENTORS

Mentors are there to help students in the process of projects or theses. Because all mentors were students once, they know very well about the problems that one might face when working on a project or thesis, especially, when it’s the first one. So take the chance to keep in touch with your mentor and consider the advices given by your mentor.

In order to work smoothly with your mentor, report regularly about the progress of your work, about what you have achieved and where you encountered problems. Plan ahead the meetings with your mentor. In particular, when you seek feedback on drafts of written work, your mentor needs to find time to read it first.

When you have questions, be specific and focused! Otherwise, you risk that the given answer will not help you much. Also avoid asking trivial technical questions, unless you have consulted all other available sources before.

Keep in mind that a mentor cannot help in all situations. In particular, mentors usually do not know the solution to the *scientific* problem to be solved (otherwise you would not be working on it). But your mentor will provide assistance on how to approach the problem and to develop the solution. After all it’s the student who has to master the task of *scientific* problem solving in order to receive a *scientific* degree.

The grade that a student receives for the submitted work depends on various factors, including the legally binding examination regulations, the complexity of the addressed *scientific* questions, the quality of the report and software, and so forth. Prof. Wolkenhauer provides a document that you might want to look at for self-evaluation (see links below).

## 1.6 SUBMISSION PROCEDURE

If not otherwise arranged with your mentor, the following things are to be submitted:

- Bound copies (to be delivered to the students office) according to the official regulations in force
- PDF copy of the entire document and additionally a TXT copy of the abstract
- Slides of project presentations and defense talk
- Original figures and screenshots in high resolution and lossless formats (e.g., PNG, SVG, OpenOffice, other vector formats)
- Software (incl. documented sources, runnable, required libraries)
- Detailed documentation (incl. compiling instructions, installation routines, tutorial explaining the basic elements of the software)

## 1.7 LINKS

- University Library
  https://www.ub.uni-rostock.de
- IEEE Digital Library
  https://ieeexplore.ieee.org
- ACM Digital Library
  https://dl.acm.org
- SpringerLink
  https://link.springer.com
- Eurographics Digital Library
  https://diglib.eg.org
- The DBLP Computer Science Bibliography
  https://dblp.uni-trier.de
- TeXstudio LaTeX Editor
  https://www.texstudio.org
- JabRef Reference Manger
  https://www.jabref.org
- Zotero Reference Manager
  https://www.zotero.org
- Mendeley Reference Manager
  https://www.mendeley.com
- 97 Things Every Programmer Should Know
  https://97-things-every-x-should-know.gitbooks.io/97-things-every-programmer-should-know/content/en/
- Study and Examination Regulations
  https://www.informatik.uni-rostock.de/studium-lehre/portalseite-studium-lehre/
- Prof. Wolkenhauer's Marking Guidelines
  https://www.sbi.uni-rostock.de/uploads/tx_templavoila/SBI_Materials_Guidelines-for-marking-reports-and-presentations.pdf

# 2 BACHELOR

## 2.1 PROJECT B.SC.

With the B.Sc. project, you develop skills in practically solving a problems in computer science, including project management, development strategies, programming, and reporting results. The project is typically team-work on a broader topic which involves specific task to be accomplished. The developed software testifies to your implementation skills, whereas the project report shows your ability to document software.

### GUIDELINES

- Thoroughly study the topic and research the literature
- Develop a general concept first
- Plan your project, define milestones, and assign tasks to team members
- Avoid reinventing the wheel, search for and use available software
- Follow well-accepted software design patterns
- Put your code under version control and commit regularly to the repository
- Document your code
- Use documentation generators to externalize your API documentation
- Test your solution with several well-defined scenarios
- Document known issues
- Meet regularly with your mentor to report your progress and discuss problems

### RULES OF THUMB

- In general, 10 pages are appropriate for the report (plus any generated API documentation), but after all it is the content that determines its quality

### ASK YOURSELF

- Does my solution solve the specified problem?
- Which design patterns have I applied?
- Can other programmers easily use and extend my solution?
- Can other programmers understand my code?
- Is my code appropriately formatted?
- Did I obey the rules and licenses of third-party software?
- Is my software free of bugs?
- Does my documentation include the programmer's as well as the end user's point of view?
- Is my application easy to use for end users?

### ALLOTTED TIME

- Allotted time is 180 hours to be expended over the course of the semester
    - Project planning and set up ≈ 8 hours (1 person-day)
    - Task and requirements analysis ≈ 16 hours (2 person-days)
    - Programming ≈ 120 hours (3 person-weeks)
    - Regular meetings ≈ 12 hours
    - Writing the report ≈ 24 hours (3 person-days)

## 2.2 BACHELOR THESIS

The bachelor thesis documents the student's ability to carry out a well-defined task following the *scientific* method. This includes a rigorous analysis of the task, a corresponding review of related work in the literature, an identification of open problems, the development of solutions to the open problems, as well as an assessment of the research results. The bachelor thesis reports on the conducted work in a *scientifically* appropriate way.

### GUIDELINES

- Analyze your task
- Define constraints for your work and requirements for your solution
- A thorough literature review that is focused on your specific task is mandatory
- Crystallize unsolved problems
- Find strategies to solve the problems
- Develop your solution on a conceptual level
- Argue for your decisions and discuss alternatives
- Avoid reinventing the wheel
- Use existing approaches and available software
- Stay focused on your task
- Read the work of former students to get accustomed to the expected structure, style, and size of bachelor theses

### RULES OF THUMB

- Do not waste your time, regularly make progress and report it to your mentor
- In general, 45 pages are appropriate, but after all it is the content that determines the quality of the thesis
- Your thesis contains about 1/3rd general material (fundamentals, literature review, etc.) and 2/3rd your own work (problem analysis, concept, implementation, etc.)
- Writing the thesis usually takes longer than students expect

### ASK YOURSELF

- What is the task to be solved?
- Do I have a sufficient understanding of the task background?
- Is my literature review comprehensive and focused on the task to be accomplished?
- Does my solution address the problems associated with the task?
- Is my solution sufficiently embedded in and contrasted against the state of the art?
- Does my thesis comply with the rules of *scientific* writing, including structure, grammar, and form?

### ALLOTTED TIME

- Allotted time is 450 hours to be expended during a period of 20 weeks
    - Task and requirements analysis, literature review ≈ 40 hours (1 person-week)
    - Concept development ≈ 120 hours (3 person-weeks)
    - Solution implementation ≈ 120 hours (3 person-weeks)
    - Writing the thesis ≈ 160 hours (4 person-weeks)
    - Defense presentation ≈ 10 hours (1.25 person-days)

# 3 MASTER

## 3.1 LITERATURE PROJECT

The literature project gives master students the opportunity to develop and strengthen their literature review skills and to acquire a good deal of background information in the area of their master thesis. You have to conduct an independent review of *scientific* literature and present the in-depth results of the review in the accompanying seminar. The insights gained are to be summarized in a survey report that also includes *scientific* questions so far unsolved in the state of the art.

### GUIDELINES

- Thoroughly review the literature
- Consult the library for books and journals on your topic
- Favor books and journal articles over conference papers, workshop papers and technical reports
- Follow the references in books and papers to find related work
- Distinguish literature that is related to your topic and work that is not
- Focus on significant results and disregard less-relevant work
- Develop an in-depth summarizing and evaluative meta-view on the topic
- Do not consider web sites as your main source of information
- Attend the seminar to see and learn how others present and discuss their work

### RULES OF THUMB

- The number of books and papers to be reviewed strongly depends on the topic and the complexity of the matter
- For the 20-minute presentation you should have no more than 20±2 slides
- In general, 30 pages are appropriate for the report, but after all it is the content that determines its quality

### ASK YOURSELF

- Does my literature review comprehensively cover the defined topic?
- Is my report focused and not excessive?
- Does my report provide an in-depth meta-view on the topic?
- Can others learn about the topic from my report without referring to the original literature?
- Did I obey the rules of *scientific* citation?
- Are my references appropriately formatted?

### ALLOTTED TIME

- Allotted time is 180 hours to be expended during a period of 15 weeks
    - Literature review ≈ 90 hours (2.25 person-weeks)
    - Writing the report ≈ 80 hours (2 person-weeks)
    - Seminar presentation ≈ 10 hours (1.25 person-days)

## 3.2 SOFTWARE PROJECT

With the software project, you improve your skills in practically solving a *scientific* problem in computer science, including project management, development strategies, programming, and reporting results. The software project defines a broader topic with a set of related problems and tasks, which are to be addressed comprehensively in a corresponding software solution. The developed software testifies to your improved implementation skills, whereas the project report shows your ability to document your solution.

### GUIDELINES

- Thoroughly study the topic and research the literature
- Develop a general concept first
- Plan your project and define milestones
- Avoid reinventing the wheel, search for and use available software
- Follow well-accepted software design patterns
- Put your code under version control and commit regularly to the repository
- Document your code
- Use documentation generators to externalize your API documentation
- Test your solution with several well-defined scenarios
- Document known issues
- Attend the seminar to see and learn how others present and discuss their work

### RULES OF THUMB

- For the 20-minute presentation you should have no more than 20±2 slides
- In general, 10 pages are appropriate for the report (plus any generated API documentation), but after all it is the content that determines its quality

### ASK YOURSELF

- Does my solution solve the specified problem?
- Which design patterns have I applied?
- Can other programmers easily use and extend my solution?
- Can other programmers understand my code?
- Is my code appropriately formatted?
- Did I obey the rules and licenses of third-party software?
- Is my software free of bugs?
- Does my documentation include the programmer's as well as the end user's point of view?
- Is my application easy to use for end users?

### ALLOTTED TIME

- Allotted time is 180 hours to be expended during a period of 15 weeks
    - Task and requirements analysis ≈ 16 hours (2 person-days)
    - Development ≈ 130 hours (3.25 person-weeks)
    - Writing the report ≈ 24 hours (3 person-days)
    - Seminar presentation ≈ 10 hours (1.25 person-weeks)

## 3.3 MASTER THESIS

The master thesis documents your ability to solve a *scientific* problem following the *scientific* method. This includes a rigorous analysis of the task, a corresponding review of related work, an identification of open problems, the development of solutions to the open problems, as well as an evaluation of the research results. The master thesis reports on the conducted work in a *scientifically* appropriate way.

### GUIDELINES

- Analyze your task
- Define constraints for your work and requirements for your solution
- A thorough literature review that is focused on your specific task is mandatory
- You may reuse results from your literature project, but do not copy-and-paste
- Crystallize unsolved problems
- Find strategies to solve the problems
- Develop your solution on a conceptual level
- Argue for your decisions and discuss alternatives
- Avoid reinventing the wheel
- Use existing approaches and available software
- Stay focused on your task
- Evaluate your results
- Read the work of former students to get accustomed to the expected structure, style, and size of master theses

### RULES OF THUMB

- Do not waste your time, this is a full-time job
- Writing the thesis usually takes longer than students expect
- In general, 80 pages are appropriate, but after all it is the content that determines the quality of the thesis
- Your thesis contains about 1/3rd general material (fundamentals, literature review, etc.) and 2/3rd your own work (problem analysis, concept, implementation, etc.)

### ASK YOURSELF

- What are the *scientific* questions to be answered?
- Do I have a sufficient understanding of the problem background?
- Is my literature review comprehensive and focused on my topic of research?
- Does my solution address the problems associated with the *scientific* questions?
- Is my solution sufficiently embedded in and contrasted against the state of the art?
- What is the *scientific* contribution of my work?
- Do I provide a convincing evaluation of my results?
- Do I obey the rules of *scientific* writing, including structure, grammar, and form?

### ALLOTTED TIME

- Allotted time is 900 hours to be expended during a period of 20 weeks
    - Task and requirements analysis, literature review ≈ 100 hours (2.5 person-weeks)
    - Concept development ≈ 240 hours (6 person-weeks)
    - Solution implementation ≈ 240 hours (6 person-weeks)
    - Evaluation ≈ 40 hours (1 person-week)
    - Writing the thesis ≈ 240 hours (6 person-weeks)
    - Defense presentation ≈ 40 hours (1 person-week)