

Software Tools Academic Writing

Part II - Lecture 6

Today's Overview

- Writing Research Papers
- Assignment Report

Tips for Academic Writing (on my homepage):
<http://www.cs.auckland.ac.nz/~lutteroth/other.html>

Writing Research Papers



Research Papers

Typical structure of a research paper (or thesis):

- 1. Title**
- 2. Abstract:** brief summary, typically 100-200 words
- 3. Introduction:** motivation, context, overview
- 4. Related Work:** what are others doing
- 5. Requirements:** What are you aiming for? Why?
- 6. Design:** the theory / our new ideas
- 7. Implementation:** how we did it
- 8. Evaluation:** why it is good/useful/better than others
- 9. Conclusion:** summing up the results

Short conference papers: typically around 4 pages

Long conference papers: typically about 10 pages

Paper Writing Strategy

- 1. Related Work:** get an overview and note down points
- 2. Requirements:** from related work or from real users
- 3. Design:** collect ideas
 1. Create structure with bullet points / mind map
 2. Create figures
- 4. Implementation:** create a prototype
 1. Start small and extend it bit by bit
 2. Experiment and collect results
(more bullet points and other data)
- 5. Evaluation:** compare and refine your work
(if necessary, go back to 2 or 3)
- 6. Title, Abstract, Introduction and Summary** can be done last

Finding Related Work

1. Gather phase
 - Keyword search (e.g. Google Scholar, ACM, IEEE)
 - Follow up the references (cited and citing papers)

1. Filter phase:
read only abstract and throw blanks out
2. Reading phase

The “someone else has already done it” problem

- Look again, is it really the same?
- Related work is good!



Writing about Related Work

- 1. Summarize** in a few bullet points what each related paper is about
 - What did they try to do? What was novel about it?
 - Did they achieve it? Did they evaluate it?
- 1. Organize** the related works by grouping them
 - Define categories, write one section per category
 - Possibly subcategories, subsections
 - E.g. categories for IDE project:
presentation integration, functional integration, process integration
 - Alternative: organize by time rather than category
- 1. Analyze & Compare**
 - What are the difference between the works?
 - Strengths? Weaknesses?

Requirements

- **What do we want? How important is it? Why?**
E.g. for IDE: code completion, integrated syntax checking, GUI builder, customizability...
- **Where from?**
 - From related work (what do others think/do?)
 - From real users (ask/survey them or read forums)
 - From real products (what do other systems do?)
 - Through analysis (what is logically required?)
- **Organize** in categories (sections and subsections)
 - Functional requirements (what does it do?)
 - Non-functional requirements (how does it do it?)
E.g. usability, performance, safety, security, ...

Design



How do you achieve your requirements?

- Explore the design space of your project analytically
- Start with an overview and then go down into the details
- What are the design alternatives?
- What are the advantages/disadvantages of each alternative?
- Which alternative do you choose and why?
- Always argue with your requirements (they are your aim)

Implementation



How did you build your system?

- What features?
- What tools/technologies were used?
- Implementation challenges and how you solved them
- Advantages and disadvantages of your implementation
- Use screenshots and/or small code snippets for illustration

Introduction



What are you doing? Why are you doing it?

1. Introduce the topic and the context

2. Motivate the research

- Interesting applications?
- Significant consequences (e.g. cheaper, faster)?

1. Research questions:

What are you trying to find out or trying to show?

2. Outline of the paper

(“Section 2 gives an overview of related work...”)

Conclusion and Abstract

Conclusion

- Sum up what it is all about
- Sum up your achievements
- Point out some future directions (e.g. new research questions)

Abstract (typically ~200 words)

- What is your project about?
- How did you do it?
- What are your results and why are they significant?

Writing Style

- **Sections:** good overall structure is the first step
- **Paragraphs:**
 - Each expresses one idea clearly
 - Split larger ones, join smaller ones
- **Sentences:**
 - Simplicity and clarity
 - Use examples for explaining complex stuff
 - Split larger ones (no runaway sentences)
- Avoid redundancy

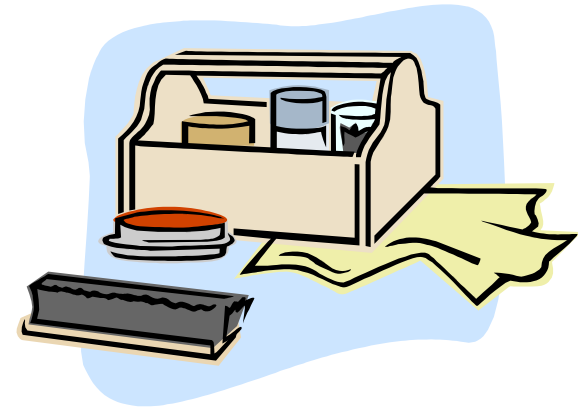
Beating Writer's Block



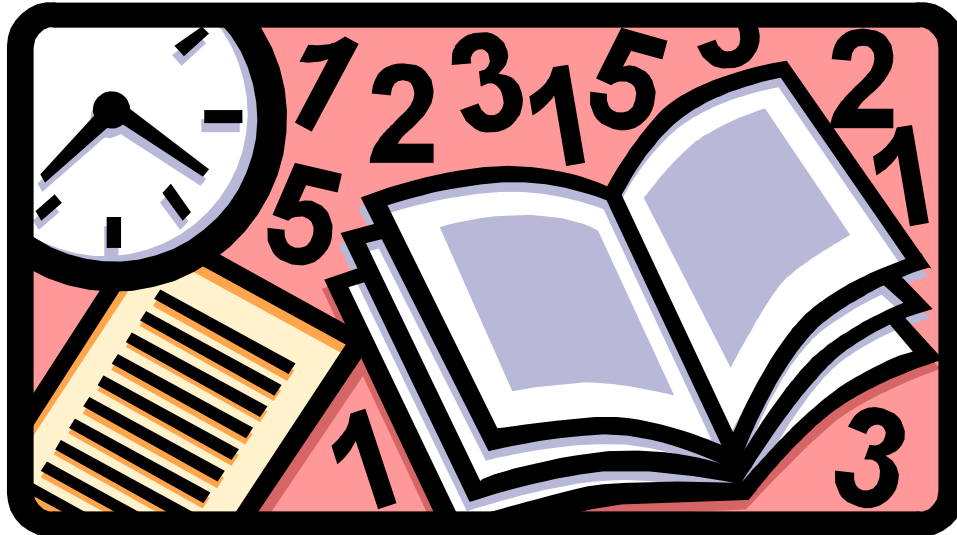
- Vary the **structure**:
 - Just write section/subsection headers
 - Just write bullet points, flesh out later
- Vary the **topic**:
 - Write about anything that comes to your mind (e.g. some related work, design, introduction, ...)
 - Organize/ reshuffle the parts later on
- Vary the **modality**:
 - **Visual**: create figures first, then simply describe what you see
 - **Auditive**: talk to others about it; write exactly as you would explain it verbally
 - **Kinesthetic**: Do some development or some experiments, then describe what you have done

Polishing your Paper

- Same as with software development: iterative and incremental refinement
- Get (early) feedback from others:
 - Is it easy to understand?
 - Spelling/grammar
 - Obvious omissions?
 - Could there be more/less figures?
 - Other interesting references?
- Emphasize your contribution (abstract, intro, conclusion)
 - How is your work different? Better?
 - How have you evaluated your work?



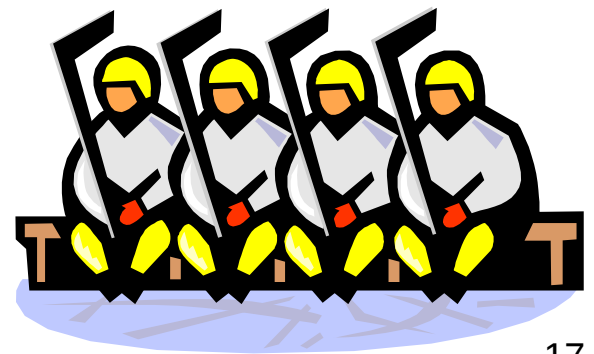
Assignment Report



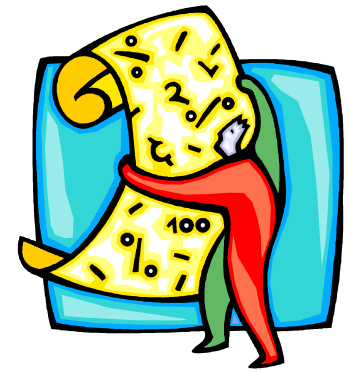
Assignment Report

25% in total for one team research project

- The project report has to be written **individually** (everything else can be teamwork)
- 5 pages IEEE style
<http://www.cs.auckland.ac.nz/~lutteroth/students/IEEETemplate.doc>
- Text, figures, bibliography
- **10 hours** for report (IEEE style)
- Will be graded by marker
- Submit as Word file
- 1st June Assignment Dropbox



Report Grading Schedule



Approx. **5 pages** (including figures) IEEE style

1. Introduction:

Have you introduced the project and its aims?

2. Related Work: Project background? Have you cited and described academic related work (≥ 4 **citations**)?

3. Requirements: What needed to be done? Why?

4. Design: How did you design your solution? Why?

5. Implementation: How did you implement it? What did you contribute? How did the team work? Challenges?

6. Conclusion: Achievements? Conclusions? Lessons? Future/unfinished work?