

## Goals

The goals of these lectures:

- describe the general research/development process
- discuss the specific requirements for a CSE thesis
- show how to go about achieving these effectively

At the end, you should ...

- understand precisely what's required of you
- produce a better result (project/report/seminar)
- use your time more effectively

Week1-1

## Topics

In these lectures, we'll talk about:

- goals/requirements of 4th-year thesis
- the process of doing research/development
- getting started on a project
- preparing/delivering the seminar
- *writing the literature review (Thesis A Report)*
- *doing and evaluating the work*
- *writing the final thesis (Thesis B Report)*

Week1-2

## Goals/Requirements of Thesis

Aims of the 4th-year thesis:

- give you exposure to research/implementation topics
- give you experience in tackling a sizeable project
- allow you to "put together" what you've learned

Requirements:

- *seminar* (project overview + plan) (in 06s2, week 12) (5%)
- *report* (literature review + plan) (due 06s2, week 12) (5%)
- *thesis* (project + evaluation) (due 07s1, week 14) (90%)

Week1-3

## Goals/Requirements of Thesis (cont.)

For definitive description of Thesis requirements:

- CSE Thesis web site<sup>1</sup>
- 4th-year Thesis Coordinator ... John Potter<sup>2</sup>

If any of these slides disagree with the above,

the above takes precedence (i.e. they are correct)

Week1-4

## FAQ

**Q:** How long should  $X$  be? ( $X \in \text{chapter, Report, Seminar, Thesis}$ )

**A:** As long as is necessary to make it convincing.

**Q:** When is  $Y$  due? ( $Y \in \text{Report, Seminar, Thesis}$ )

**A:** Check [www.cse.unsw.edu.au/thesis](http://www.cse.unsw.edu.au/thesis)

**Q:** What happens if I can't finish?

**A:** You get less marks than you would if you finished.

**Q:** Can I get an extension?

**A:** No ... if Thesis A is late/missing, you get AF.  
If Thesis B is late, you suffer heavy *late penalties*.

Week1-5

## FAQ (cont.)

**Q:** What must I do to get good marks?

**A:** Depends on whether you're asking supervisor or assessor

Supervisor:

- knows everything you did (or ought to)
- can assess based on continuous performance

Assessor:

- (most likely) sees only Seminar, Demo, Thesis
- can assess based on what she/he observes in S/D/T

To be safe: ask what they're looking for.

Week1-6

### Why a Thesis is not an Assignment

A thesis is significantly different from an assignment:

- it is, typically, *open-ended*
  - there is not an obvious single “correct” answer
  - you have more say in the direction the work takes
- it has a much longer time-frame
  - you need more *self-discipline* to get things done
  - you have more responsibility to *plan* your progress

If you're still in “assignment mode”, break the task into 2-week-long steps and treat each one as an assignment (but, alas, no late penalty).

Week1-7

### Different Types of Theses

Theses have been classified into:

- |     |   |
|-----|---|
| RES | carry out a small focussed piece of research  |
| DEV | build a software and/or hardware system   |
| R&D | combination of the above two ...<br>build a system, but needs research to get it done |

Expectations for each type are slightly different (see later)

Week1-8

### What is Research?

Activity that advances the store of human knowledge, e.g.

- explaining a previously unexplained phenomenon
- developing techniques to do things better
- proposing a new viewpoint on a cultural system
- proving a theorem

Scientific/engineering research follows the “scientific method”:

- Observe ... Hypothesize ... Test ... (repeat)

Week1-9

### What is Research? (cont.)

**Example:** astronomer discovering rotating star pairs

Observation:

- individual stars that appear to pulsate/change  
(changing brightness, different spectra at different times, ...)

Hypothesis:

- such “stars” are actually rotating pairs (intuition)

Test:

- examine stars using different instruments  
(e.g. very powerful optical observation to reveal individual stars)

Week1-10

### Computing Research

Writing a piece of software, no matter how complex, isn't generally regarded as research in itself.

However, it would be considered research if

- it uses a new method/algorithm/data structure  
(the new method must be *demonstrably* better than earlier approaches)
- it solves a problem not previously solved by computer  
(will typically involve development of new methods, anyway)

Week1-11

### Computing Research (cont.)

**Example:** devising a new database indexing method

Observation:

- queries of type Y are not efficient with existing methods
- try to figure out what it is about Y queries that makes them slow

Hypothesis:

- my new method (X-trees) will handle Y queries *better*

Test:

- analyse computational complexity of X-trees for Y queries
- build an X-trees indexing library, test on a variety of data

Week1-12

### Computing Research (cont.)

Evaluation of computing research:

- solves existing problem more effectively than before
- solves a wider range of problems than before (generalises)

Demonstrations of effectiveness follow two tracks ...

Theoretical, e.g.

- analyse complexity, prove upper/lower bounds, ...

Experimental, e.g.

- build prototype; measure performance on range of data

Week1-13

### What is Development?

Construction of artefacts/systems to solve problems.

Implicit notion that new systems are *better* than existing.

There is still a (minor) research aspect:

- you must demonstrate effectiveness
- you must analyse and describe limitations

A very important requirement for a DEV project:

- follow a well-defined project development methodology
- document the development process in the thesis

(cf. research projects: the result is paramount, method matters less)

Week1-14

### Computing Development

Aim: build a system to meet a demand or solve a problem.

May involve developing software, hardware, or a combination.

The goal is clearly to build the system, but you must also:

- follow a (software) engineering methodology (+ document it)
- provide a demonstration that the system *works effectively*
- note any unsolved problems and limitations

Week1-15

### Computing Development (cont.)

Characteristics of effectiveness are problem-dependent, e.g.

- time and space, for most computing projects
- bandwidth utilisation, for networks
- usability, if there's a user interface
- accuracy (precision/recall), for web search engines

Must use *appropriate* evaluation instrument for project.

Must be honest about effectiveness (even negative results are useful).

Week1-16

### Doing Research

The process is an elaboration of "observe/hypothesis/test":

- understand the broad topic area
- establish an evaluation framework (issues, metrics)
- look at what others have done and evaluate
- find an area of "weakness" in existing work
- devise a (better) solution
- evaluate the effectiveness of the solution

Hint: document all parts of this process **as they are done**.

Week1-17

### Reporting Research

Reporting follows, more or less, the steps in the process:

- |                     |                                       |
|---------------------|---------------------------------------|
| <b>Introduction</b> | sell the topic, summarise aims        |
| <b>Background</b>   | set the context, review literature    |
| <b>Own Work</b>     | what have you done, exactly           |
| <b>Evaluation</b>   | convince us that it's good            |
| <b>Conclusion</b>   | summarise achievements (and failures) |

Week1-18

## Reporting Research (cont.)

### Introduction:

- outline the topic area, significance, originality, ...
- give overall aims of your work, summarise contribution

### Background:

- describe the problem you are trying to solve in detail
- establish evaluation framework (what makes a *good* solution?)
- describe *and analyse* what others have done already

Week1-19

## Reporting Research (cont.)

### Own Work:

- describe your proposed method/approach in detail

### Evaluation: (experimental)

- describe evaluation process (what are you measuring, how, why)
- report *and discuss* results of evaluation

### Conclusion:

- summarise what you achieved
- *and* what you didn't achieve ... and suggest how to fix it

Week1-20

## Doing Development

Has similarities to research ... but like "observe/build/test":

- understand/refine the requirements
- establish an evaluation framework
- look at what others have done and evaluate
- look at what methods are available, and choose
- devise a solution (implement system)
- evaluate the effectiveness of the solution

Hint: document all parts of this process **as they are done**.

Week1-21

## Reporting Development

Reporting follows, more or less, the steps in the process:

<b>Introduction</b>	sell the topic, summarise aims
<b>Background</b>	set context, evaluate approaches
<b>Own Work</b>	what have you done, exactly
<b>Evaluation</b>	convince us that it's good
<b>Conclusion</b>	summarise achievements (and failures)

Week1-22

## Reporting Development (cont.)

### Introduction:

- outline the topic area, significance, benefits ...
- give overall aims of the project

### Background:

- describe the problem you are trying to solve in detail
- establish evaluation framework (how to recognise a good solution)
- describe *and analyse* what others have done already (maybe no-one has attempted this before?)
- evaluate possible implementation methods/approaches

Week1-23

## Reporting Development (cont.)

### Own Work:

- describe implementation process (SE) and final product
- if it has a user interface, give a tour of this

### Evaluation:

- describe evaluation process (what are you measuring, how, why)
- report *and discuss* results of evaluation

### Conclusion:

- summarise what you achieved
- *and* what you didn't achieve ... and suggest how to fix it

Week1-24

## Getting Started

Week1-25

## Getting Started

What you should be doing as soon as you have a topic:

- *think* about the topic ... understand all aspects, etc.
- find out what others have done before (reading required)  
( Google<sup>3</sup> ... Citeseer<sup>4</sup> ... DBLP<sup>5</sup> )
- consider what's need to evaluate your work  
(don't bother considering evaluation-by-bluff ... it doesn't work)
- make notes on everything that you look at/think about
- set up a thesis web site? (for holding the notes)
- meet your supervisor and find out what they expect
- establish a consistent Thesis working pattern (weekly meeting?)

**Do all of this** before week 6 and assignments get heavy.

Week1-26

## Getting Started (cont.)

Things you should do **today** ...

- create a directory called *thesis* (or *4910* or ...)
- create subdirectories: *report*, *seminar*, *thesis*
- if doing implementation, create a directory called *system*
- create subdirectories:
  - *notes*: where you type up ideas *as you think of them*
  - *papers*: to keep electronic copies of reference material
- if keen, set up a web site (e.g. Wiki) for your thesis work

You should do this *today* because you have no assignments to do yet.

Week1-27

## Thesis A

Thesis A aims for you to demonstrate that ...

- you have a thorough understanding of the topic
- you have identified an aspect that requires work
- you have an approach for solving the problem  
(and you can argue the likely effectiveness of this approach)
- you have a plan for carrying out the work  
(including time-frames for tasks, knowing how to evaluate, ...)

Week1-28

## Thesis A Seminar

Week1-29

## Seminar Structure

- |                      |  |
|----------------------|--|
| <b>Introduction</b>  | sell the topic, summarise aims<br>(5-6 minutes)        |
| <b>Background</b>    | set context, evaluate previous work<br>(12-15 minutes) |
| <b>Proposal/Plan</b> | how do you plan to tackle the problem<br>(2-3 minutes) |
| <b>Bibliography</b>  | give references for all work cited<br>(as you go)      |

Seminar = summary of Report, publicity for project, chance for feedback

Week1-30

## Seminar

The seminar aims to:

- give you a chance to practice your presentation skills
- let you show that you have met the goals of Thesis A
  - convince others that you're studying an important/interesting problem
  - demonstrate that you've done some research/thinking about it already
  - have a plan for the rest of the year to solve the problem

If you already have some results to show, that's a bonus.

Target your seminar at fellow thesis students *except* target the hard-core technical stuff at your supervisor and assessor.

Week1-31

## Seminar (cont.)

Suggested structure for presentation:

1. Title slide (name of project, your name, ...)
2. Example to illustrate the problem (1-3 slides)
3. Introduction to problem; statement of goal (1 slide)
4. More detailed problem description (3-4 slides)
5. Survey/critique of existing approaches (3-4 slides)
6. Outline your approach to solve it (1-3 slides)
7. Plan for your work for the rest of the year (1 slide)

Use max 15 slides; you cannot cover more in 25 mins.

(The most difficult part of preparing a presentation is deciding what to leave out)

*Practice* the talk to a friend (for timing and explanation clarity).

Week1-32

## URLs

1. [www.cse.unsw.edu.au/thesis/](http://www.cse.unsw.edu.au/thesis/)
2. [www.cse.unsw.edu.au/info/potter.html](http://www.cse.unsw.edu.au/info/potter.html)
3. [www.google.com.au/](http://www.google.com.au/)
4. [citeseer.csail.mit.edu/](http://citeseer.csail.mit.edu/)
5. [www.informatik.uni-trier.de/~ley/db/](http://www.informatik.uni-trier.de/~ley/db/)

Week1-33