

CSW

Computer Science Writing

Fiona Polack

# Who's here, doing what?

- All 3Y0s (MEng):
  - assessment, lectures, practicals
- All those taking PR3 (third year project):
  - lectures

# AIMS

- To prepare for project work
- To prepare for writing up and assessment
- To prepare for project presentation
  - This applies to MEng students only

# Learning Outcomes

- To be able to make an informed choice of text preparation facilities, and use these to present written text clearly and accurately
- To understand the difference between draft and final copy
- To be able to use literature effectively, including appropriate citation styles and constructing a critique
- To prepare and give a presentation on an academic subject, in a fixed time

# DISCLAIMER

- This is a new module and a new venture
- If you don't like it
  - Or think it should be different
    - Say so... I may be able to modify later sessions!

# Lectures

10 lectures:

- Thursdays, 1615, B/B/006, weeks 1-2
- Mondays, 1615, P/L/001, weeks 2-4
- Tuesdays, 1215, P/L/001, weeks 2-4
- Thursdays, 1615, P/X/001 , weeks 3-4

**All attend the lectures**

# “Guest” lectures

- Preparing a project using MS Word,
  - Alistair Edwards, Week 2 Thursday
- Preparing a project using latex
  - Jeremy Jacob, Week 3 Monday
- Scientific Method
  - Susan Stepney, Week 3 Thursday
  - may also be Week 4 Monday

# Practicals I

## **MEng Students Only**

Practical slots for practice presentations

- Week 5 only (2-8 November)
  - Monday 1315 G/045
  - Monday 1615 A/EW/102
  - Tuesday 1415 G/045
  - Tuesday 1715 A/EW/102
  - Thursday 1315 A/EW/106



# Practicals II

## **MEng Students Only**

- 8 or 9 students; 5 minutes each
  - Timing strictly enforced
- Feedback from demonstrators
- Provisional schedule
  - <http://www-course.cs.york.ac.uk/csw/MEngsByName.htm>
  - <http://www-course.cs.york.ac.uk/csw/MEngsBySession.htm>
- **TELL ME IF YOU'RE NOT ON THE LIST  
OR WOULD LIKE TO SWAP**

# Practicals III

## **MEng Students Only**

- Each student prepares a 5-minute (unassessed) presentation.
- presented to the other students in the session
- A demonstrator will supervise each session
  - Hope to use PR5 presentation feedback forms
- Details on web
  - <http://www-course.cs.york.ac.uk/csw/Presentation.html>

# A silly interlude

If you are an MEng third year

- Stand on the right

If you are doing a PR3 project

- Stand on the left

Anyone else: come to the front

**MOVE NOW**

# AND THEN....

- Find one or two other people from your side
- Go and sit down together (so you can talk)
- Discuss projects whilst the others get sorted
- Elect a spokesperson

We'll use group discussion from time to time

- MEngs talk about presentation subjects!

# WHAT IS A PROJECT?

Discuss what might determine the nature  
of projects in the Department

2-3 minutes: timer

What do you think....?

# Project definitions: rationale I

- Most degrees are accredited with IEE/BCS.
- Projects must conform to the Engineering Council definitions.
- Ours are a lot bigger than the minimum!
  - <http://www.bcs.org/BCS/Products/HEAccreditation/courseguidelines.htm>

# Project definitions: rationale II

- The National “Honours” and “Masters” level degree designations also affect project definitions.
  - <http://www.qaa.ac.uk/crntwork/nqf/ewni2001/contents.htm>



# Project definitions

- All the project definitions are linked to the Projects web pages:
  - <http://www.cs.york.ac.uk/projects/>
  - <http://www-users.cs.york.ac.uk/~fiona/PWG/>
- Check the definition ... your supervisor may not have done so!

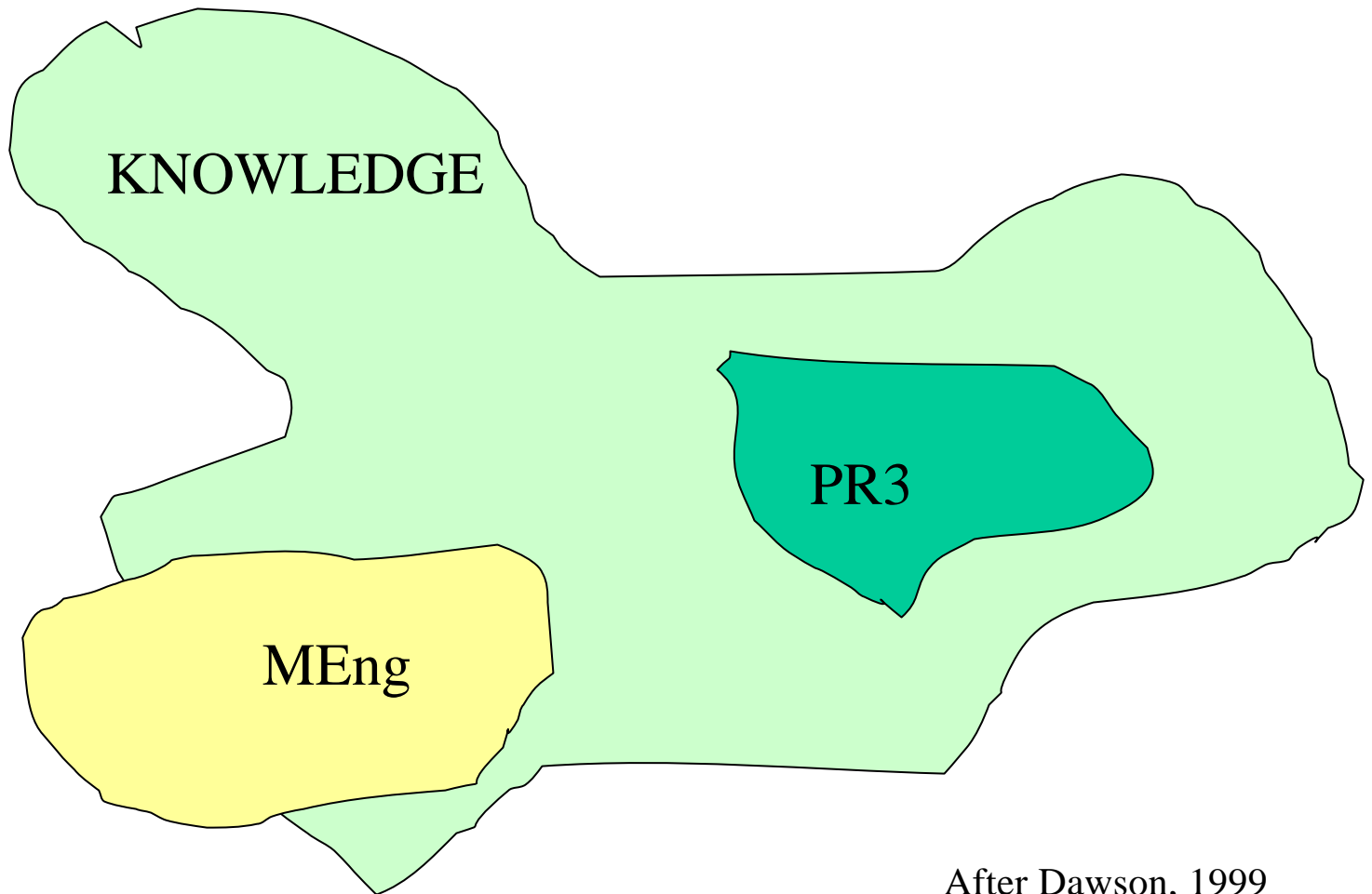
# MEng Project Definition

- MEng projects (PR5) are Masters Level,
  - the degree qualifies for CEng exemptions
- A research or engineering project - the approach is similar
- There must be an element of originality

# PR3 Project Definition

- Third year projects (PR3) are Honours level
  - all treated as if they were part of the IEng accreditation
- They are engineering projects

# Projects and Knowledge



After Dawson, 1999

# “Engineering”

Discuss what Engineering means

2-3 minutes: timer

# Definition I

The Engineering Council (SARTOR 1997), an engineer is:

*“...one who has and uses scientific, technical and other skills to create, enhance, operate or maintain safe, efficient systems, structures, machine, plant, processes or devices of practical and economic value.”*

# DEFINITION II

The Engineering Council is less exact these days:

- [http://www.ukspec.org.uk/definitions\\_CEEng\\_IEng\\_EngTech.asp](http://www.ukspec.org.uk/definitions_CEEng_IEng_EngTech.asp) defines CEng, IEng and all the other levels of engineer

# What is an Engineering Project?

An engineering-style project *report*

- Follows a lifecycle and a method
- Aims to “build” something
- States and explains the method
- Evaluates the method and product

Any project can be written up in this format.



# General Engineering Issues

Professional organisations require,

- Quality
- Reliability
- Timeliness
- Maintainability

Think what these might mean for your project,  
and strive to achieve them!

# Engineering Lifecycle

Project undertaken and *written up* in terms of:

- Requirements: context, constraints
- Design: may be several stages
- Build: software, hardware, proof, experiment...
- Evaluate: product, method, results: relate to requirements

These are good practice in all projects.

# Method

Any method will do (if it's appropriate)

- Software Engineering methods - agile programming, RAD, unified process etc;
- Hardware methods for specification/design;
- Research method: problem-hypothesis-experiment
- Algorithm design/proof: problem-plan-doing it

# STATE THE METHOD USED

Even if it was a retro-fit method  
(if there's more than one method, state  
them all)

# Fitting a Project to a Lifecycle

- Design-and-build projects should fit ok
- Other projects:
  - Explain the goals, constraints, givens, literature
  - Write up the method: hypothesis, experimental design, plan etc
  - Write up and evaluate the results
  - Consider what maintenance might mean!

# And MEng Projects...

- All the above apply
- More thoughtful and advanced in approach and content....
  - Systematic, comprehensive understanding
  - Critical awareness and evaluation
  - Forefront of the field; original knowledge or application

# PLANNING

# Planning I

Some supervisors ask for a formal plan

- Planning is always a good idea

Why?



# Planning II

Because things go wrong in projects

- If you don't know how long it should have taken, you don't know how to revise your plan

Plans are made to be revised!

# HOW DO YOU PLAN A PROJECT?

Discuss what is important...

2-3 minutes: timer

# Planning a Plan I

- How do you like to work?
  - Concentrated blocks or parallel to other things?
  - Department, Library, at home, in a café?
- What facilities do you need, and when?
- How easily do you write (long reports!)?
  - Do you need a proof reader?
  - Do you write it all in one go or in stages?

# Planning a Plan II

- What needs doing? All projects have
  - Literature review: context, resources etc
  - Design or set-up or planning
  - Implementing or doing
  - Results or testing
  - Evaluation
- Schedule at least these activities

# Planning a Plan III

- Reports are about writing, not just doing
- Decide, and plan in, whether you write up
  - at the end (inadvisable)
  - in stages
  - as you go along
- Plan in at least a week at the end to finish the report

# Planning: advice

Anything that can go wrong will go wrong

– (source anyone?)

- If you write up at the end, it **WILL** take at least 3 weeks
- However well you plan and work, something **WILL** go wrong in the last month/week/day/hour....

# Doing the Work

- A less crucial part of the project... so long as it gets done
- Scale the work to fit the time
- The key things you need are
  - Clear notes
  - Good justification or motivation of what you did

# The Report and its focii



# The hourglass model

- Start work & report with a wide focus
  - Lit. review explores this focus and context
- From lit. review, develop & justify sharper focus
  - Design to the sharper focus
- Implement & test what you can of the design
- Evaluate against to the original context

# The hourglass report

- Adapt the report title to what you did
  - project title may not be ideal for report
- Adapt the report introduction and conclusion to the rest of the report
- Make the developing focus clear at the start/end of each relevant chapter

# The hourglass model example

## *DB transaction project I*

Project: modelling database transactions

- Method: an incremental development
  - Review, plan,
  - implement on case study,
  - plan next.
- Lit. review:
  - Existing modelling approaches
  - Definitions related to db transactions

# The hourglass model example

## *DB transaction project II*

- Use Lit. to select a particular aspect
  - eg specification using UML; integrity at physical design;... or whatever
- Design (for example...)
  - Language extension for the chosen aspect
  - Metamodel (language definition)

# The hourglass model example

## *DB transaction project III*

- Implementation (for example...)
  - Apply to a case study that highlights the chosen aspect of transaction modelling
  - Use the designed language on the case study
- Note what does and does not work

# The hourglass model example

## *DB transaction project IV*

Evaluation questions:

- Did the language & metamodel suffice?
- How did the result compare with existing models?
- How would the new work fit into existing methods?
- Was the approach to the project adequate?

What if your Project  
“doesn’t work”?

# No project “fails”

(though students sometimes do!)

- The plan can be revised
- The proposal can be rewritten, revetted etc
- The report reflects what was possible
  - Evaluation may be easier than for a project that achieved its original goal
- In extremis, put in an EC or apply for an extension...
  - But don't risk relying on these to help you!



# Reasons for “Failure”

- Method proved inappropriate
- Resources late or not available
  - Includes not enough time
- Research proved project infeasible or already done

# Method proved inappropriate

- Can you change methods part way through?
  - use the experience to enhance the method justification;
  - Admit what happened and exploit it!
- Otherwise
  - Write up the project as done
  - In evaluation explain why method was inappropriate and what would have been better

# Resources late or not available

Necessary hardware, software not available,

OR

Your time ran out

OR

There was more work than anticipated

- Adapt the project to fit what could be done
  - Adapt plan and report
- Comment (neutrally) on the problems in the Introduction and evaluation

# Project infeasible or already done

- Consider adjusting the project
  - May redraft the proposal
  - Write the report on the revised project
    - you don't need to explain why the original was not done
- Vary the context to an area not already been done
  - Perhaps use lit. review to reflect on the existing work
  - requirements/design for revised context

# Report presentation issues

# WHAT STYLE SHOULD YOU USE FOR A PROJECT?

Discuss what is important...

2-3 minutes: timer

# General Style I

It's a report, not a manual.

- Write for an academic audience
  - Succinct
  - Precise
- Support points with evidence
  - From literature
  - From your research

# General Style II

- Use the marking criteria
- Marking forms are on line:
  - PR3: <http://www.cs.york.ac.uk/projects/BEng.pdf>
  - PR5 (MEng) <http://www.cs.york.ac.uk/projects/MEng.pdf>
- The general marking criteria are on line:
  - [http://www-users.cs.york.ac.uk/~fiona/PROJECTS\\_WEBPAGES/markin\\_g\\_form.html](http://www-users.cs.york.ac.uk/~fiona/PROJECTS_WEBPAGES/markin_g_form.html)



# Some Good Projects I

- Sedding (2004) PR3
  - Engineering presentation of research
  - not particularly “literary”, but well structured and very clearly written
  - Nice use of literature, and good referencing
  - [http://www.cs.york.ac.uk/library/onlineprojlib/project\\_files/2004/ug/jps118/jps118\\_project.pdf](http://www.cs.york.ac.uk/library/onlineprojlib/project_files/2004/ug/jps118/jps118_project.pdf)

# Some Good Projects II

- Goude (2003) PR3
  - Experimental project; could have been structured better as an engineering report
  - Very clearly written
  - Excellent evaluation chapter, and in-text justification or rationale
  - [http://www.cs.york.ac.uk/library/onlineprojlib/project\\_files/2004/ug/jps118/jps118\\_project.pdf](http://www.cs.york.ac.uk/library/onlineprojlib/project_files/2004/ug/jps118/jps118_project.pdf)

# Some Good Projects III

- Crawford (2004) PR3
  - Well-presented engineering and methods
  - Good use of sources in literature and development
  - [http://www.cs.york.ac.uk/library/onlineprojlib/project\\_files/2004/ug/adc109/adc109\\_project.pdf](http://www.cs.york.ac.uk/library/onlineprojlib/project_files/2004/ug/adc109/adc109_project.pdf)

# And some other names...

- PR3
  - Weeks, Gurney (PR3 2004)
  - Thorn, Lloyd (PR3 2003)
- PR5 (MEng final year)
  - Naylor, Whitham, Davis, W. (PR5 2004)
- PR4 (MMath final year):
  - Walker (PR4, 2004), Minichiello (PR4 2003)

# Audience

- Assume a general computer science background.
- Elaborate specialist aspects of the project area
  - In literature review (with references)
  - In other chapters
  - In appendix if necessary

# Readable reports

- Don't leave the reader guessing
  - Say what you're going to discuss,
  - discuss it
  - Summarise it
- Applies at report, chapter, and section level
- This is different to a lab book (or a novel!), where the plot unfolds as you progress

# Linguistic issues

# Language

These are academic norms, but not essential:

- Third person passive
  - “The project presents....”
  - “No further examples were found...”
- Present tense for main text
  - “This section contains....”
  - “The design uses UML diagrams....”



# Drafting the report

- Everyone writes draft first
- native *and* non-native English speakers make mistakes and typographic errors
- Spelling checkers do NOT proof read
  - Do spell check, but also read it carefully

# Proof readers

- If you want your report checking, you may ask someone to *proof-read* it

BUT:

- Don't expect your supervisor to do it
- Don't ask a student from this Department;
- Don't accept ANY technical input (risk of collusion).

# Final Copy

- Clear language
- No typographic errors or wrong spellings
- Clear structure, with “waymarks” to content
- Consistent citation
- Consistent referencing of diagrams, tables, appendices etc
- All chapters and sections included

# Report don'ts

- Don't "tart up" the text
  - but do lay it out nicely, with section headings, bullets etc
- Don't try to "sell" your product
- Don't embellish the truth (or the product)
- Don't repeat substance (but do provide clear "waymarks" through the text)

# REPORT CONTENTS I

Preamble and Appendices

# Preamble

Included in word & page counts:

- Title page
- Abstract
- Contents page(s)
- Acknowledgements

Be canny – follow the rules in the Handbook, but don't put in more than you need

# Appendices (Marked)

Included in word & page counts and listed on the title page as part of these counts

- Appendices that you want marking, eg
  - Large diagrams, tables, results which are essential but peripheral to the main text
- Use appendices - to make the text of the report flow well

# Appendices (unmarked)

- Supplementary material for the reader
- Early versions of diagrams, if you wish
- Code listings and implementation detail
- Screen shots (if not an HCI project!)

On title page, list the unmarked appendices and state that these are excluded from the word & page counts



# REPORT CONTENTS II

Context and literature review

# Context

- A project always has a context
  - Those things that motivate, limit and define the project from without
- Context needs to be explored in the early chapters of the report
  - PR3, PR5 etc refer to all the contextual material as literature review

# Literature review as context

Thus, for projects, *literature review* is not only a review of relevant literature

- It could cover any context-setting material
  - Selection of methods, based on literature
  - Review of what has been done in the area
  - Review of things from other areas that are relevant to the project

# Project Literature review

## Literature Review

- May be one chapter or several
- May include sections of other chapters
  - For instance, each design/implement/evaluation chapter could include justifications of approach or content based on cited sources

# Bibliography

- Between main text and appendices
- NOT included in page and word counts
  - There's no penalty for
    - Citing lots of sources
    - Giving full citations
    - Other good bibliographic practices

Good use of literature always  
benefits a project.

WHAT CAN YOU USE?

HOW CAN YOU FIND  
MATERIAL?

Discuss possible sources...

2-3 minutes: timer

# Finding Material

- [google](#)
- [citeseer](#)
- [University library Catalogue](#)
  - Current periodicals, Interlibrary loans
- Pages of interested academics
- Anything else you can think of....



# Books

- Good for background material
- Good for information about established methods, notations etc
- Often out of date (even new ones)
- Not usually sufficient on their own

# Journals

- Academic papers, reviewed and accepted by an academic community
- May include seminal or key works
- Usually reasonably accurate
- Most journals take over a year to publish a paper, so not really state of the art

# Conferences

- Proceedings of academic talking-shops
- State of the art, both good and bad
- Reviewed and accepted by academic panels, but discrimination may be poor
- Good for review purposes, but expect to find contentious points or errors

# LNCS, LNAI etc

- <http://www.springeronline.com/sgw/cda/frontpage/0,11855,3-164-12-72397-0,00.html>
- Specialist Computer Science publishers
- Many conference proceedings
- Some specialist issues
- Find them in the library and browse them!

# Technical reports

- From educational and industrial organisations
- “unpublishable” reports e.g.
  - Pre-publication papers and results
  - Highly technical
  - Case studies
- If relevant, use “published” version
- [eg. http://www.cs.york.ac.uk/ftpdir/](http://www.cs.york.ac.uk/ftpdir/)

# Web pages

- Technical information, manuals, user hints etc on commercial products
- Non-commercial software
- Shortcut to published material
  - cite the published source where possible
- A lot of rubbish appears on the web
- Don't worry if a page disappears
  - cite the link and the date you last consulted it

# Popular press

- Reputable magazines such as *Nature*, *Science* and *New Scientist* may provide general context
- Sometimes good for evidence of trends
- In general, too sensationalist for serious literature review
  - Back up popular press citations with serious academic references

# Unpublished sources

- Draft papers, comments, emails
- In general, get originator's permission to cite
- If cited, be clear that it is *unpublished*
- Use *personal communication* footnotes if necessary<sup>1,2</sup>

1. Personal communication, F. A. C. Polack, 14 October 2004
2. Email, Prof. Ding Bats, received 21 May 2003, used with permission



# Reading and noting

- Always record the details of the source when you get it
  - Saves a huge amount of time later
- Note comments as well as content
  - Review is about critique, not just content
- Be realistic about the amount of review you can do

# Citation to distinguish your Contribution

The report must make clear what's yours & what is other people's work

- Reference all use of others' material, such as,
  - Background literature
  - Method steps,
  - Equations,
  - Reused code or structures etc

# Academic culture, Citation & Plagiarism avoidance

# Prof. S. Stepney, PD1 lectures, 2003-04

- [http://www-  
course.cs.york.ac.uk/csw/ExtractSSPD1Ethics.p  
pt](http://www-course.cs.york.ac.uk/csw/ExtractSSPD1Ethics.ppt)

# Plagiarism checking

- The Department is exploring plagiarism detection
- Electronic copies of the project may be checked through automatic detection tools
  - Selected projects (at random)
  - Ones that raise markers' suspicions

**You have been warned!**

# What is the purpose of a Bibliography?

To show that you understand the context of your work.

To allow other people to follow up your references:

- For their own research
- To check your facts

# Bibliography rationale

- Cite things used so that others could find them
- The Student Handbook indicates that common knowledge does not need to be cited
  - You don't have to cite the O.E.D. every time you use an English word
  - You don't have to cite e.g. Wirth every time you mention Pascal, or Knuth every time you mention LaTeX
- In most other cases, it's safer to cite

# Citation styles

- Some use *[Stepney03]* or *[Ste03]* in text
  - May need *[Ste03a]* and *[Ste03b]* etc
- In bibliography:  
*[Stepney03] S. Stepney, Lecture Notes, PD1 Ethics, 2003*
- Others use numbers: *[1]* in the text
- In bibliography:  
*[1] S. Stepney, Lecture Notes, PD1 Ethics, 2003*

Either is fine, but be consistent



# Fine-grained citation

- Computer Scientists are slapdash about citing precisely
- If you want to use a finer grain of citation:
  - In text: [1, chapter 2]; [Stepney03, p3]
  - In bibliography, don't repeat the entry
  - just one entry for every reference to [1] or [Stepney03], as before.

# Citation Planning

- Latex bibtex does most of the layout for you (see J. Jacob's material)
- Microsoft Word has cross-referencing, which does some of it for you (see A. Edward's material)
- Always check the bibliography for accuracy

# Bibliographic content

Enough information to trace your source:

- Books – author(s) or editor; title; publication date
- Journals – article author and title; journal title, volume and number (as appropriate)
- Web pages – full http address; date accessed

# Bibliographic entry examples I

- C. J. Date, *Introduction to Databases*, 2000
  - Could add edition (7<sup>th</sup> edition)
- F. Polack, A Case Study using Lightweight Formalism to Review an Information System Specification , *Software Practice & Experience*, 31(8), 2001
  - Could add pages (pp57-80), month (July)

# Bibliographic entry examples II

- Jill Srivatanakul, John A Clark and Fiona Polack, Security Zonal Analysis, Technical Report YCS-2004-374, University of York Computer Science Department, 2004.
- <http://www-course.cs.york.ac.uk/pd1/ethics.pdf>, last accessed 20 September 2004
  - Could add author (S. Stepney) and subject (PD1 notes on Ethics).

# Some things to avoid in citation

- Don't use footnotes for referencing
- Don't put more than one entry per source in the bibliography
- Don't include uncited material in the bibliography
  - use a separate listing of uncited but influential works if necessary

# Using Literature

# Critical Review

- Projects get credit for justification, motivation, explanation
- Review contributes to these if it is
- *Critique* means commentary
  - Positive and negative, and neutral
- Don't just regurgitate published material



# Good and bad reviews

- The best reviews present an argument or discussion, supported by literature
- The worst reviews just present the contents of what was read with no comment
- You get no credit (and may lose it) if a review does not include appropriate citations

Giving a presentation  
**MEng only**

# THE PRESENTATION

- Marked presentations were first used for MSc students in September 2004.
- PR5 presentation marked by
  - report second marker
  - and someone else (not supervisor)
- The PR5 mark combines the presentation mark and the report mark in the ratio 5:95.

# PR5 Presentation details I

- A 10-minute talk, followed by questions
  - Timing strictly enforced
- 10 minutes is not very long:
  - keep it simple
- Sessions running in parallel

# PR5 Presentation facilities

- A standard dual-boot PC
  - Networked to the student system
- Projector connection for own computers
- Overhead projector and flipchart
- Student wireless network access

# PR5: Personal Computers

You may use your own portable computer, but

- All electrical equipment must be power checked in advance
- Wireless access requires advance registration

Contact [support@cs.york.ac.uk](mailto:support@cs.york.ac.uk) for advice

- Don't leave it to the last minute

# PR5 Presentations including a hardware demo

- Specialist requirements must be notified
  - at least 14 days before the report hand-in
- Presentations may take place at separate times and locations

It's your duty to notify the presentations co-ordinator of your requirements

# PR5 Presentation contents

1. Project title and objectives
2. Context
  - quick summary of background and literature
3. How you went about it (method)
4. What you achieved
  - you must include a demo if you can
5. Evaluation – how you did it and what you concluded



# PR5 FEEDBACK

- Presentation feedback will be available with the returned copy of the report
- Currently, there is no formal feedback on the project; if this changes, you'll get that too!

# CSW Presentation I

- Week 5 sessions
- Review one recent paper by a member of staff (see module webpage)
- Use this structure:
  1. Paper title and subject area
  2. Context: general CS background
  3. What the paper covers
  4. Evaluation – what are its merits and demerits?

# CSW Presentation II

- 5 minute presentations in flat classrooms
- Facilities are limited, and so is time
  - Ambitious presentations are not acceptable
  - Computer presentations may not work
  - OHP slides, hand written, are fine
  - Black- or white-board should be provided
- Don't waffle