Coursework commentary 2015–2016

CO3320 Final project report

General remarks

Students are reminded that the individual Project course is the most important in your degree; it can carry considerable weight with potential employers, who typically place a high value on self-managed problem solving and the results obtained. Many students report that they have been able to shine at job interviews by referring to the work they carried out for their Project.

There are many different kinds of Projects, and many different kinds of achievements. However, all of the very best Project report submissions share some of the characteristics listed below, which have been described and discussed in previous years' Reports.

Exemplary, First class project work

In this section, we explain the characteristics found in most exemplary Projects; those that tend to achieve a First class mark.

Clear literature review: It is important to have a clear literature review that explains the connection your work has to related work in the field. It should explore the context of previous attempts to solve problems related to the one you are tackling. The best kind of literature review is one that establishes exactly what the current state-of-the-art offers in terms of effectiveness, efficiency and usefulness. The literature review sets the scene for your Project to make an original contribution.

Results: A Project worthy of a first class mark will typically achieve results at least as good as the state-of-the-art, and for higher marks above 80 per cent there should be some contribution beyond the state-of-the-art. Students who have provided a literature review that clearly establishes where the state-of-the-art lies can demonstrate that their results have **improved** on it.

Publishable work – the ultimate accolade: The very best Project submissions contain work that is publishable. Such Reports typically achieve marks in the high 80s and even 90s. In order to be publishable, a Project report would need to both advance the state-of-the-art, and also present compelling scientific evidence to **demonstrate** that the student's contribution has gone beyond the state-of-the-art.

The importance of evaluation: The very best Project submissions had an extremely thorough and compelling evaluation section. Such an evaluation will typically present results for the performance of the proposed approach (which would have been carefully implemented and compared with current techniques available). A good evaluation will compare against multiple different alternatives, and will attempt to set demanding goals for itself.

Methods: Projects that contain a very strong evaluation, by definition also contain methods and results that are sufficiently robust to support the evaluation. Marks are awarded for a systematic approach, and also for the quality of the documentation that explains the implementation. Many students lose marks unnecessarily because although they may have

implemented a system, the evaluation and documentation was insufficient for the examiners to be clear about what had been achieved. Students should ensure that they allow sufficient time for the write-up of their Project as well as for the implementation and groundwork required to collect data upon which they make scientific claims in the Report.

Testing versus evaluation: It is important to understand the difference between testing and evaluation. A good Project needs both. Testing establishes that the implementation performs as expected. Essentially, it aims to answer the question: 'Is the system correct?'. However, an evaluation goes beyond this by asking about the performance of the system. Typically, the system has been implemented in order to evaluate a particular claim to a scientific contribution. It is the scientific contribution that is addressed during the evaluation stage of the Project.

Sufficiency: In this section we moved to the other end of the spectrum to consider what is required in order for a Project to pass; just clearing the bar for sufficient contribution for a pass mark.

Some evidence of a contribution: Projects that barely contain enough information to pass, may contain **sufficient** information to just scrape past the threshold. Such projects may have minimal evaluation and very little testing, but in order to pass a Project does need to make **some** contribution.

As in previous years, this year (2015–16) saw several Projects that were nicely presented, and contained a reasonable account of the literature on the topic they chose to study. However, they made no contribution beyond this and therefore did not meet the threshold required for a pass.

To pass, the Project has to make a contribution – albeit a modest one – in order to simply cross the threshold for a pass. This contribution might take the form of a rudimentary implementation, perhaps of a prototype and not necessarily a fully working system. In order to achieve a pass, the Project also needs to contain sufficient documentation in order to clearly state what has been implemented and achieved.

Non-implementation based Pass: Projects can achieve a pass without having an implementation of a system, but in this case the report has to include some other kind of contribution. Students often choose to include either a questionnaire or some form of interview/survey. They should be aware that such a Project **can** obtain the highest marks if done exceptionally well. However, examiners are often disappointed that students who chose to follow this path tended to arrive at a very substandard Project report. The Project is the capstone of your degree, and should build on what you've learned in the taught courses; namely, it should focus on some aspect of computing.

In order to do exceptionally well with a Project aimed at answering a question through a survey of stakeholder, a questionnaire or interviews, you need to follow best scientific practice in the conduct of the interviews/ questionnaire, and the evaluation of the scientific contributions.

Many Projects in this genre simply report raw statistics, but do not attempt to evaluate any scientific hypotheses. Such Projects could be improved by including more detailed statistical analysis and a clear understanding of the scientific hypotheses being investigated. Necessarily, such projects would require some understanding and deployment of relevant statistical analysis techniques for investigation of hypotheses.

Students who simply include a well-written literature survey and rudimentary contributions in terms of either a prototype implementation or a survey with only shallow analysis, typically achieve a mark between a mid-Third class mark and a mid-Upper second. The precise mark will often depend upon completeness, and the quality of presentation and adherence to standards as set out in the subject guide.

Pitfalls to avoid in your Project report

You can further improve your mark by paying attention to the following pitfalls, which were found to be prevalent this year, and which tend to contribute to a reduction in the marks awarded.

1. **Avoid relying on code listings**. This means the inclusion of code in the Report without proper formatting, documentation and layout. This year, many students chose to include large chunks of unformatted and largely uninteresting program code. Examiners tend to ignore program code **unless you make a clear effort to explain which parts of the code relate to the design, and how they do so**. You are warned that you may lose marks for padding out your Project report with unnecessary information, and that one such form of padding consists of unnecessary inclusion of program listings. It is much better to focus on specific aspects of the system, and use fragments of code embedded in the natural language narrative in the body of your report to explain the implementation; long listings of code are uninteresting and **detract** from the Project.
2. **Clearly delineate your own code contributions**. It is essential to indicate which parts of the code have been written by you and which parts have been reused from other sources. Reusing code is **not** deprecated, but clarity about which code is reused and which code is original to you is essential.
3. **Narrative explanation is essential**. The inclusion of design documentation requires adequate textual explanation. UML diagrams on their own are insufficient to describe the design of a system.
4. **Avoid sloppiness**: This includes sloppy presentation of references, with missing information and/or messy layout. Such examples of sloppy scholarship often catch the eye of the examiner and cause a reduction in marks awarded. In extreme cases, you may be reported for plagiarism. Avoiding this pitfall takes a relatively small amount time.