Individual Project Guide to Deliverables

6CCS3PRJ

1 Introduction

The individual project is the most important single piece of work in the degree programme. It provides you with the opportunity to: engage in a detailed investigation of an issue in computer science, to demonstrate independent study skills and original thought, to plan and organise a large piece of work over a long period of time, to put into practice some of the ideas and methods that you have been taught throughout your degree programme, and to explore novel forms of computer science tools and techniques. Whatever your level of academic achievement so far, the purpose of the project work is for you to demonstrate your individuality and inspiration in tackling a significant issue in computer science.

The final year project is a substantial exercise in the application of computer science principles, and should involve some or all of the following: research, specification, design, implementation, validation, and critical analysis. The final year project will provide you with experience of managing your own work and will require you to produce a substantial piece of written work, and a significant piece of software.

The project module is especially important for providing you with a range of transferable skills that you are expected to exhibit in your work. For example, you are expected to: critically evaluate relevant material, provide evidence of an ability to construct logical and reasoned arguments that are supported by evidence, to manage your own studies (including working to deadlines), to communicate your ideas orally and in writing, to reflect critically on your own work, and to exhibit an awareness of professional issues (see below). These skills will be assessed in the project module (cf. the Project Assessment Guidelines) and they should be used by you as a checklist to apply to the work that you produce. If you are unsure about any aspects of these transferable skills and how they relate to your project work then you should ask your supervisor for further advice. An important transferable skill is being able to formulate requests for help.

There are eight deliverables for the project: the final report and the oral examination plus five project progress report presentations and a background and specification report. All of these submissions are considered when deciding the final mark to be awarded for a piece of project work.

2 Progress Reports

You will have five group supervision sessions. For each session, you will be asked to present the current state of your project (what you have achieved since the last meeting or since you have been allocated your project), any barriers that are currently causing you problems progressing with your project, and your plan of work for time until the next supervision meeting. You will need to prepare a slide deck of three slides, each presenting one of these three points. Your slide deck needs to be submitted on KEATS at least one day before your group-supervision meeting and no later than the deadline indicated on the project schedule.

In your first group supervision meeting, you will need to demonstrate that you have clearly defined the problem space for your project. You should discuss the motivation and the scope of the project, list its aims and objectives and include a short bibliography of relevant literature. You should state what software platform (programming language(s), software packages, and software environment), hardware platform and operating system(s) you will be using. You should also include a detailed plan of progress for the remainder of your project (both terms). This must consist of a Gantt chart and a list of deliverables. When you submit your first progress report slides, you should also state if ethical clearance (also known as ethical approval) is needed for your project (please discuss this with your supervisor, you can get more information about ethical clearance here).

3 Background and Specification Progress Report

By the end of the first term you are expected to have a draft of the early part of your final report, the Background and Specification Progress Report (BSPR). This report should include the following things:

1. Background and context for the project
2. Review of any relevant literature, including web sites
3. Requirements
4. Specification
5. Design

The BSPR should set the project into context by motivating the project and relating it to existing published work, which you will have read at the start of the project when your approach and methods were being considered. There are usually many ways of solving a given problem, and you shouldn't just pick one at random. Describe and evaluate as many alternative approaches as is reasonable.

The published work considered in the report may be in the form of research papers, articles, text books, technical manuals, or even existing software or hardware of which you have had hands-on experience. Don't be afraid to acknowledge the sources of your inspiration; you are expected to have seen and thought about other people's ideas; your contribution might be to extend existing work or to apply existing ideas in novel ways. Whatever you choose as the focus of your project, you must be sure to avoid plagiarism. If you take another person's work as your own and do not cite your sources of information and inspiration then you are being dishonest; in other words you are cheating. When referring to other pieces of work, cite the sources at the point where you refer to or use them, rather than just listing them at the end.

Where appropriate, the BSPR should give a detailed, complete specification of any code to be implemented for the project. This may be based on material in the first progress report but should extend it. You should aim for your specification to be as low-level and detailed as you can manage at the time you submit the report. You should use (as appropriate) standard software engineering methods for specification, such as: requirement lists, use cases, state machines, entity relationship diagrams, and UML notations.

The Background and Specification Progress Report should be about 10 pages long but can be longer if necessary and can even be shorter if appropriate. Your supervisor can advise you on this.

As with other project deliverables, you must submit (upload to KEATS) your BSPR by 4pm on the submission date. As the BSPR is a more substantial report than the other progress reports that you submit,

your supervisor will need time to consider your work. Therefore, your supervisor will read and will provide you with comments on your BSPR at some point soon after the January examinations have been sat. Please note that if you fail to submit your BSPR on time then your supervisor is not obliged to provide comments on your BSPR submission.

4 Final Report

The final project report is the most important aspect of the project. The project report serves to show what you have achieved and should demonstrate that:

- You understand the wider context of computing by relating your choice of project, and the approach you take, to existing products or research.
- You can apply the theoretical and practical techniques taught in your degree programme to produce a significant piece of written work and a substantial piece of software that addresses a particular problem in computer science.
- You are capable of objectively criticising your own work, discussing its implications, and making constructive suggestions for improvements or further work based on your experiences so far.
- As a computing professional, you can explain your thinking and working processes clearly and concisely to third parties who may not be experts in the field in which you are working.

Most of the project assessors will not have followed the project throughout and will only have a short time to listen to a presentation or see a demonstration. For this reason they will rely heavily on the report to judge the quality of your work. Moreover, if your overall degree mark places you at a boundary between two degree classifications, the final award decision can be influenced significantly by the quality of your project report. You should appreciate that the external examiners, who play a crucial role in the final recommendation, have only the report by which to judge your project performance; the same applies to projects that are put forward for a prize.

Many students underestimate the importance of the report and make the mistake of thinking that top marks can be achieved simply for producing a good piece of software. This is fundamentally not the case. Many projects have been graded well below their potential because of an indifferent or poor write-up. In order to get the balance right you should consider that the aim of the project is to produce a good report and that software, hardware, theory, etc. that you developed during the project are merely a means to this end. Do not make the mistake of leaving the write-up to the last minute. Ideally, you should produce the bulk of the report as you go along and use the last week or two to generate the final form.

The layout and formatting of the report is important but often neglected. A tidy, well laid out and consistently formatted document makes for easier reading and is suggestive of a careful and professional attitude towards its preparation. Conversely, a poorly structured project report detracts from the overall quality of the submission and that will be reflected in the mark that you are awarded.

It should be obvious that quantity does not automatically guarantee quality. Although it is important to appreciate that the appropriate size and structure of a report can vary significantly from one project to the next, the body of the project report must not exceed **25,000 words** (N.B. the body of the report does not include the program listings and other material that may be included in the appendices).

Any project such that the main body of the report exceeds the equivalent of 25,000 words will be deemed to have not satisfied the presentational requirements for an individual final year project. In

this case, the project work will be awarded a zero mark. Additional material may be included in appendices but please note that the examiners are not obliged to read this additional material. The length of the main body of the final report is expected to be considerably less than 25,000 words. Conciseness, clarity and elegance are invaluable qualities in report writing, just as they are in programming, and will be rewarded appropriately. Conversely, a failure to express yourself succinctly is a shortcoming that, if exhibited, will be reflected negatively in the mark that you are awarded.

5 Final Report Structure

In general terms, you should try to ensure that your report contains the following elements (the exact structure is, of course, up to you):

5.1 Title page

This should include the project title and the name of the author of the report. You can also include the name of your supervisor if you wish.

5.2 Originality avowal

(See also Program listings below)

The first page of the report after the Title page should contain the following statement certifying the work as your own: ``I verify that I am the sole author of this report, except where explicitly stated to the contrary.'' Your signature and the date should follow this statement.

5.3 Abstract

The abstract is a very brief summary of the report's contents. It should be about half-a-page long. Somebody unfamiliar with your project should have a good idea of what your work is about by reading the abstract alone.

Acknowledgements: It is usual to thank those individuals who have provided particularly useful assistance, technical or otherwise, during your project. Your supervisor will obviously be pleased to be acknowledged as he or she will have invested quite a lot of time overseeing your progress.

5.4 Contents Page

This should list the main chapters and (sub)sections of your report. Choose self-explanatory chapter and section titles. You should include page numbers indicating where each chapter and section begins. Try to avoid too many levels of subheading. In particular, stick to sections and subsections; subsubsections are usually avoidable.

5.5 Introduction

This is one of the most important components of the report. It should begin with a clear statement of what the project is about so that the nature and scope of the project can be understood by a lay

reader. It should summarise everything that you set out to achieve, provide a clear summary of the project's background and relevance to other work; describe the economic, commercial, customer and/or research context of your project; and give pointers to the remaining sections of the report, which will contain the bulk of the technical material.

5.6 Background

The background should set the project into context by motivating the subject matter and relating it to existing published work. The background will include a critical evaluation of the existing literature in the area in which your project work is based and should lead the reader to understand how your work is motivated by and related to existing work.

5.7 Body of report

The central part of the report usually consists of three or four chapters detailing the technical work undertaken during the project. The structure of these chapters is highly project dependent. They can reflect the chronological development of the project, e.g. design, implementation, experimentation, optimisation, evaluation, etc (although this is not always the best approach). However you choose to structure this part of the report, you should make it clear how you arrived at your chosen approach in preference to other alternatives. In terms of the software that you produce, you should describe and justify the design of your programs at some high level, e.g. using OMT, Z, VDL, etc., and you should document any interesting problems with, or features of, your implementation. Integration, verification and testing are also important to discuss in some cases. Finally, you should discuss aspects as the future maintenance of the developed software. You may include fragments of your source code in the main body of the report to illustrate points; the full source code is included in an appendix to your written report (see below).

5.8 Evaluation

Be warned that many projects fall down through poor evaluation. Simply building a system and documenting its design and functionality is not enough to gain a good mark. It is extremely important that you evaluate what you have done both in absolute terms and in comparison with existing techniques, software, hardware, etc. This might involve quantitative evaluation, for example based on performance measures, or something more qualitative, such as functionality or ease-of-use. It may also involve a discussion of the strengths and weaknesses of your work. Avoid statements like "The project has been a complete success and all of the problems in this area of computer science are now solved "- you will be shot down immediately! It is important to understand that there is no such thing as a perfect project. Even the very best pieces of work have their limitations and you are expected to provide a proper critical appraisal of what you have done. Your assessors are bound to spot the limitations of your work and you are expected to be able to do the same.

5.9 Legal, Social, Ethical and Professional Issues

Your report should include a chapter with a reasoned discussion about legal, social ethical and professional issues within the context of your project problem. You should also demonstrate that you are aware of the Code of Conduct & Code of Good Practice issued by the British Computer Society and

have applied their principles, where appropriate, as you carried out your project. You could consider aspects like: the effects of your project on the public well-being, security, software trustworthiness and risks, Intellectual Property and related issues, etc.

5.10 Conclusions and Future Work

The project's conclusions should list the key things that have been learnt as a consequence of engaging in your project work. For example, "The use of overloading in C++ provides a very elegant mechanism for transparent parallelisation of sequential programs", or "The overheads of linear-time n-body algorithms makes them computationally less efficient than O(n log n) algorithms for systems with less than 100000 particles". Avoid tedious personal reflections like "I learned a lot about C++ programming...", or "Simulating colliding galaxies can be real fun...". It is common to finish the report by listing ways in which the project can be taken further. This might, for example, be a plan for turning a piece of software or hardware into a marketable product, a set of ideas for possibly turning your project into an MPhil or PhD, or a discussion of how the project could accommodate future requirements or extensions.

5.11 Bibliography

This consists of a list of all the books, articles, manuals, etc. that are referred to in the report. You should provide enough information to allow the reader to find the source. You should give the full title and author and should state where it is published, including full issue number and date, and page numbers where necessary. In the case of a text book you should quote the name of the publisher as well as the author(s). The College's guidelines on acceptable citation can be found here.

5.12 Appendix

The appendices contain information that is peripheral to the main body of the report. Information typically included in the Appendix are things like tables, proofs, graphs, test cases or any other material that would break up the theme of the text if it appeared in the body of the report. It is necessary to include your source code listings in an appendix that is separate from the body of your written report (see the information on Program Listings below).

5.13 User Guide

You might have to provide an adequate user guide for your software at the discretion of your supervisor. The guide should provide easily understood instructions on how to use your software. A particularly useful approach is to treat the user guide as a walk-through of a typical session, or set of sessions, which collectively display all of the features of your package. Technical details of how the package works are rarely required. Keep the guide concise and simple. The extensive use of diagrams, illustrating the package in action, can often be particularly helpful. The user guide is sometimes included as a chapter in the main body of the report, but is often better included in an appendix to the main report.

5.14 Program Listings

Complete source code listings must be submitted as an appendix to the report. The project source codes are usually spread out over several files/units. You should try to help the reader to navigate through your source code by providing a ``table of contents'' (titles of these files/units and one line descriptions). The first page of the program listings folder must contain the following statement certifying the work as your own: ``I verify that I am the sole author of the programs contained in this folder, except where explicitly stated to the contrary''. Your (typed) signature and the date should follow this statement.

All work on programs must stop once the code is submitted. You are required to keep safely several copies of this version of the program. Your examiners may ask to see the last-modified dates of your program files, and may ask you to demonstrate that the program files you use in the project examination are identical to the program files you have submitted. Any attempt to demonstrate code that is not included in your submitted source listings is an attempt to cheat; any such attempt will be reported to the KCL Misconduct Committee.

6 Submission of Final Report

The final report is to be submitted electronically via KEATS. You should make sure in advance that you can upload your report so that there are no last minute problems. The final report must be uploaded on KEATS by **midnight** on the due date (do not attempt to submit your final report to your supervisor). Note that unless there are compelling reasons for excusal, late submissions of the final project report will receive a mark of zero.

There are final reports of the previous year's projects available for consulting in the Important Information section.

7 Important notes

Students are required to demonstrate that all of their products meet the highest professional standards possible. They should demonstrate that they have considered, addressed and critically assessed ethical, economic, cultural, legal and environmental issues of relevance to their project work. Students must also demonstrate that in their decision making they have:

- Given sufficient attention to ethical codes and ethical principles, taken account of safety, security and privacy requirements, appreciated human and personal rights, respected relevant laws and standards, exhibited an understanding of the effects of project decisions on communities, the environment and individuals. All project students are strongly encouraged to read BCS material on professional issues which can be accessed here.
- Given the importance of the project in the degree programme, the penalties for plagiarising project work are especially severe (and include the possibility of permanent exclusion from the college with no possibility of receiving a KCL degree). The department has considerable expertise in detecting plagiarised work. As a student at King's you will have read the College's statement on plagiarism and you will have already signed a declaration to state that you understand the term and agree to abide by the statement on plagiarism. The KCL statement on academic honesty and integrity can be found here. The college's guidelines on avoiding plagiarism and acceptable citation formats can be found here and here. If you require further explanation of the College's policy on plagiarism then please ask your supervisor for guidance.

8 Useful references

- The Essence of Computing Projects - A Student's Guide by Christian W. Dawson. Published 2000 by Prentice Hall.
- Project-Based Software Engineering (An Object-oriented Approach) by Evelyn Stiller and Cathie Le Blanc. Published 2002 by Addison Wesley.

Document history

Date	Responsible	Comment
September 2017	Natalia Criado	Initial document version
06 September 2019	Steffen Zschaler	Updated information on progress reports
24 August 2020	Steffen Zschaler	Removed reference to engineering projects