

# CSC3002 Computer Science Project

## Student Handbook 2016-17

**IMPORTANT: This is a double module worth 40 CATS points; normally one of the conditions for the award of a BEng/BSc degree in Computer Science is that CSC3002 is passed.**

### ChangeLog

23 September 2016	Published on QOL
13 October 2016	Deadline for Initial Project Description and Work Plan to be emailed to supervisor extended.
27 <sup>th</sup> October	Fix cross referencing error for deadline for Initial Project Description and Work Plan – it wasn't updating later in the document

### 1. Important Dates (all subject to change)

<b>Project Selection by</b>	Wednesday 5 <sup>th</sup> October 2016, 4pm
<b>Initial Project Description and Work Plan emailed to supervisor by</b>	Friday 28 <sup>th</sup> October 2016, 4pm
<b>Problem Description, Solution Approach and Work Plan to main office by</b>	Friday 9 <sup>th</sup> December 2016 4pm
<b>Interim Demo to supervisor completed by</b>	Wednesday 14 <sup>th</sup> December 2016
<b>Final Dissertation Handed in to main office and submitted to Turnitin by</b>	Tuesday 9 <sup>th</sup> May 2017 4 pm
<b>Final Dissertation submitted to Turnitin by</b>	Wednesday 10 <sup>th</sup> May 2017 4pm
<b>Demonstration and Expo Day</b>	Thursday 11 <sup>th</sup> May 2017

### 2. Module Co-ordinator

The module coordinator is Dr Des Greer. Email: [des.greer@qub.ac.uk](mailto:des.greer@qub.ac.uk)

### 3. Computer Science projects

The CS Project requires the construction of a software system in order to demonstrate your ability to apply practical and analytical skills present in the Computer Science pathway as a whole.

On completion of this module, the successful student should demonstrate competency in the techniques necessary for a rigorous and disciplined approach to software construction. Typically, this involves

- Analysing a problem, collecting information about the problem, considering possible solutions and being innovative in developing a solution approach
- Develop a high quality solution and critically evaluate that solution

Projects are arranged with members of staff as supervisors. You will be allocated a supervisor near the start of the term. Computer science projects are completed as individual projects. Typically, projects are set in

an area of interest to the member of staff's research, but alternatives can be arranged with individual members of staff. ALL projects must be approved in advance by the allocated supervisor.

The project is a means of utilising the skills and knowledge you have acquired from the modules that you have studied (as well as, possibly, your professional experience). Coding is only one aspect of a project! Problems that on the surface seem relatively straightforward often have hidden depths. Successful projects require organisational as well as technical skills. You may have to develop new skills and gain new knowledge in the project and the assessors should take this into account when marking.

#### **4. Selection of project**

A list of project areas will be published during the first week of term. Projects available will have a brief description of the area to be investigated and/or the software to be developed and other relevant information. Students will create a shortlist of 5 projects and then rank their project preference by Wednesday 5th October 2016, 4pm. A URL link will be sent to by email for you to complete this online. The allocation will be done shortly after the selection deadline.

Please note that there is no guarantee that you will be allocated a project in the area of your choice (because of supervisor availability constraints).

#### **5. Self-defined projects**

Defining your own project is possible, but you must find a member of staff who is willing to supervise it. The project must be something you haven't worked on already and you must adhere to the rules of normal projects and the constraints for final year projects. If you wish to work with an outside company, be aware that this is often difficult, since the needs of the company may not match those of the final year project. The needs of the company are secondary and they should be made aware of this. Entrepreneurial projects (you may want to look at <https://www.checkout.com/scholarship/>) are also possible but again you will need to find a willing supervisor and the needs and constraints of CSC3002 are the priority. In all cases the work must be new and the supervisor can insist on a given direction in the project. If you have arranged a project with a supervisor, please get the supervisor to email the module coordinator and await approval before proceeding (des.greer@qub.ac.uk).

#### **6. Course Organisation and Timetabling 2016/2017**

The dates and deadlines for the various stages of the project are given at the top of this document

Deadline extensions are not normally granted (e.g. a 2 day absence due to illness in the 1<sup>st</sup> semester will not result in an automatic extension of two days to the dissertation submission date). There is only one initial lecture in semester one and if needed another one in semester 2. All necessary documentation will be posted on QOL.

#### **7. Supervisor**

Your main point of contact for project advice is your project supervisor. You should arrange a meeting with your supervisor and agree a project with them as soon as possible.

## 8. Course Background

- The Computer Science Project is a project-based course. There is no examination. The work involved with a CSC3002 project should be similar to the work associated with two conventional modules (including time for examination preparation).
- Marks are allocated for project work submitted on or before the dissertation deadline.
- Project work is of special interest to employers and is often discussed during interviews. Project work is often cited in academic references.
- Group work is **not** permitted.

## 9. Resubmission of Projects

In the event that you do not submit or do not pass CSC3002, you will have the opportunity to resubmit your project work in August 2017. It is your responsibility to initiate contact with your supervisor, as soon as possible after the examination results are published, in order to discuss how your work needs to be revised.

## 10. Project Roles

- Student: The student is responsible for ensuring that the project is carried out effectively and completed on schedule. A student should spend on average six and a half hours per week on project work. It is the student's responsibility to record minutes of project meetings. There is a template for this on QOL. It is the responsibility of the student, not the supervisor, to ensure that his/her project is completed satisfactorily. It is the responsibility of the student to arrange regular meetings with the supervisor during the project, and record minutes of meetings.
- Supervisor: The supervisor is normally responsible for providing technical advice and guidance to the student.

If you have a query regarding a technical aspect of a project, then it should be discussed with your supervisor. If you have a non-technical problem, then it should be discussed firstly with the supervisor and then with the Module Co-ordinator. Additional technical support may be available for some projects depending upon the hardware and software used. **The staff providing this assistance must not be used as surrogate supervisors.** You are expected to communicate with your supervisor and the Module Co-ordinator via e-mail as well as face to face.

## 11. Organisation

Planning is critical to the success of the project. Plans help you determine the scope of the project, track its progress and to support decisions. The plan does not have to be right first time, but should be corrected as you learn during the project. In any case, you should devise a plan (e.g. Gantt chart or table) which outlines major predicted milestones and deliverables from the start. This should be kept updated as the project progresses and the products evolve. It is your responsibility to organise meetings with your supervisor to

keep him informed of progress. The choice of software process is open to discussion with your supervisor. Some projects suit a linear process better, while others will need to accommodate change and learning as the project progresses. Experience shows that delivering valuable functionality in the project early in semester 1 and continually tends to indicate a good project outcome in semester 2. Students who do not demonstrate steady and consistent progress throughout the period of the project often produce inadequate deliverables.

The total amount of time available for completing the project and the associated dissertation is approximately six months. In order to complete the dissertation on time, the dissertation may need to be started in parallel with the later stages of the development work. You should aim to complete all development work on the project before the end of March.

## **12. Problem Description (Friday 28<sup>th</sup> October 2016, 4pm)**

Once you have been allocated a project, you have to develop an initial problem description and work plan. This should give a good overview of the problem to be solved. Goals and requirements should be stated clearly. A description of a software / hardware development environment may be given. Verifiable criteria against which the success of the project is to be judged should be identified (e.g. features of the software and experimental results). You may state these as acceptance tests, if preferred. A Gantt chart or table which outlines the expected project development plan should be given (with major milestones and deliverables highlighted). Appropriate references should be provided.

## **13. GitLab**

You are expected to use the school GitLab facility to store your code and other artefacts as you develop them. Access details about this via the EECS Self-Service web pages. Additionally, you must agree that staff have access to the project repositories, to view, to be able to clone them for private browsing and to view and summarise statistics on the repository activity and content.

It is the student's responsibility to become familiar with using git. Guidance can be found at <https://selfservice.eecs.qub.ac.uk/docs/git/>.

At the submission deadline your access to the repository will be read only so that you can't make any more changes to code while it is being assessed.

Please make sure you regularly commit code (and other artefacts as they are generated) via GitLab. Staff may monitor your activity to ensure progress is being made.

## **14. Project Description, Work Plan and Interim System Demonstration**

As the first semester progresses and you learn more about the problem area, you should update your problem description and work plan document. This should then be handed in as a stapled document to the main office in the Computer Science building by **Friday 9th December 2016 4pm**. Before **Wednesday 14th December 2016**, you must arrange an interim demonstration in conjunction with your supervisor. At

the demonstration you should illustrate how much progress you have made with respect to a subset of requirements. These should be agreed with your supervisor at the outset of the project and only changed by agreement. Typically, an interim demonstration consists of the execution of a significant part of your system and may include initial experimental results. Having some mocked up some screens or some minor development work is not considered acceptable. There should be value in the system

### **15. Project System Demonstration**

Each student will be required to give a demonstration of their completed system to their Supervisor and to their Assessor on Thursday 11th May 2017 (demonstration day). The supervisor and assessor will assign a mark to the system produced, based on how it has met the project goals and on the quality of the product. If a system is incomplete but partially demonstrable then marks may be awarded for the completed portion of the system. Local employers from industry may attend system demonstrations. Demonstrations provide an opportunity for you to bring your work and skills to the attention of employers.

### **16. Dissertation Structure**

The dissertation should normally have the following structure. There may be circumstances where a variation in the structure is appropriate; such changes should always be discussed with your supervisor.

---

#### **Title Page**

This should have the following format:

Title of Dissertation  
A dissertation submitted in partial fulfilment of  
the requirements for the degree of  
**BACHELOR OF *SCIENCE/ENGINEERING*\*** in Computer Science  
in  
The Queen's University of Belfast  
by  
'Student Name'  
'Date of Submission'

\* as appropriate

#### **Declaration Cover Sheet (Declaration of Academic Integrity)**

Attached to the end of this document

#### **Acknowledgements**

To those who have helped the author during the project and the preparation of the dissertation and to anybody who has given financial support.

## **Abstract**

A summary (100 words) which provides an outline of the subject matter and the results, findings and/or conclusions of the dissertation.

## **Contents**

A complete list of chapters, sections, appendices etc. with page numbers.

## **Main Text**

The main body of the dissertation as described below organised as a sequence of chapters each normally containing several sections. The main text should not normally exceed 40 pages (it may be less).

## **References**

A list of references to documents (books, papers, web pages etc.) which are referred to in the main body of the text. Use the IEEE citation style as detailed here <https://www.ieee.org/documents/ieeecitationref.pdf>

**The first citation should be the URL to the software code repository which should contain the code and any other resource required to run the software.**

## **Appendices**

These should include as appropriate:

- (a) A User manual giving details on how to use the software, including details of input data, output formats and error messages.
- (b) Test results, if appropriate.
- (c) Other information which is not convenient or appropriate to include in the main body of the dissertation.
- (d) Minutes of the Project meetings.

## **The Main Text**

**The Examiners will be looking for quality rather than quantity in your dissertation.** You should try to keep the main text of your dissertation as concise as possible. Spelling should be correct, sentences grammatical, and formulae, figures and tables accurate. All figures, tables and appendices should be given numbers and headings. Your writing should be precise, concise and fluent. Avoid the first person (i.e. say what was done, rather than that you did it). Some dissertations may be organised differently from that

headings given below. In particular, some parts might have more emphasis than others for some projects. Further advice will be available from your supervisor. Below is a sample of what might be expected.

### **1.0 Introduction and Problem Specification**

Background material should be given which introduces the problem area, its context and background. You should identify the particular problem under consideration along with information about the problem area that enables the reader to understand the problem scope and nature. If your project involves a particular domain, algorithm, method theory etc., you may describe it in the introduction (alternatively or additionally, it may be described later, if appropriate). For best marks the student should show that they have systematically researched and fully analysed the problem, synthesising the relevant information.

### **2.0 System Requirements Specification**

You should provide a precise description of the system developed. These may have been written much earlier but for the dissertation they should be updated to match the final system delivered. You should list any assumptions made about the problem and any system constraints. Overall your requirements, functional AND non-functional should be complete, clear, accurate, feasible and objectively verifiable. Content depends on your project but could include:

- A complete set of function definitions (as use cases if preferred), as far as possible written so as to be testable
- Measurable and testable non-functional requirements
- Description of interfaces required such as with other software or systems
- Any specific user interface requirement
- User characteristics

The target to aim for here is that your requirements could be the basis for a contract or handing to external developers to complete.

### 3. Design

This section should describe the design of your proposed system. Normally this several parts, depending on your project:

- (i) Architectural Description of the system – textual and/or diagrammatic. This could be a simple diagram showing the components and how they relate or it could describe the choice of architectural style or pattern used.
- (ii) User Interface Design (if applicable). Show sketches of the design or screenshots with explanations of choices made, if necessary.
- (iii) Software System Design.

The role of each component and the interfaces between components should be described. There should be a clear correlation between your design and your specification.

The design should be linked to requirements and, where applicable give a critical discussion of key design decisions/styles/patterns used. There might be a data model, a UI design, details of external interfaces, and of other important issues e.g. concurrency, event handling, error and exception handling, security, data persistence. No particular notation or tool is mandated.

### 4. Implementation and Testing

You should describe any languages, packages, and libraries etc. that are used in the development of your system. There is no need to describe your code in detail. You may highlight data types and implementation techniques that are of special interest. If appropriate, you may provide:

- (a) Choice of implementation language(s)/ development environment(s)
- (b) Use of software libraries;
- (c) Key implementation decisions
- (d) A description of how some important functions and algorithms were implemented.
- (e) A description of how each component is implemented.
- (f) Discussion of Test Approach e.g. unit testing, system testing, regression testing etc; Test cases described; Testing tools used. Evidence that testing coverage was complete.

Program code can be accessed by the assessors via the git repository **so there is no need to print code listings**. It is recommended that you comment code appropriately. Programs should be written in a clear style with good program structure and well-defined data structures. The program code should reflect its design.

### 5. System Evaluation and Experimental Results

Different projects will have a different emphasis. In all cases you are expected to provide empirical results and to draw conclusions from those results. You may use your software to generate experimental results. Be sure to describe the methodology of your evaluation or experimentation. An experiment is typically described in terms of its goals, the hypotheses being tested, the subject of the experiment, what is being



measured and what is controlled, the results obtained and the analysis and interpretation of those results. Alternatively, you can assess the product in terms of how it compares with other similar products and/or in terms of user feedback (e.g. via a survey) or some measurable quality aspect such performance efficiency or reliability. Your supervisor can guide you on what is appropriate, but typically the very best projects have shown results that could be publishable with little or no work or show an exemplary empirically based evaluation of a software product

## **6. Conclusion**

A general summary evaluation of the success of the project should be given with respect to criteria identified in the introduction. A discussion of the significance of your experimental results may be appropriate. Do they agree with other previous work or ideas? How does your system compare with similar ones? An evaluation of the hardware/software environment and language used may be presented, if appropriate. Draw conclusions on the process used in the project as well. What went well? What did not go well? What are the strengths of your solution or conclusions? What are the weaknesses? Suggestions for further work should also be discussed. You can be critical and draw a negative conclusion. Not all projects will be successful. A well-explained failure is as an acceptable an outcome as a spectacular success. Assessors are looking for excellence in a critical appraisal of the work and a convincing argument for the significance of contribution in the context of wider work.

Any publication of results of the student's work is left to the discretion of the supervisor.

---

## **17. Copyright of Material Supplied**

Material supplied at the outset or during the project may be the subject of copyright (or other industrial property rights) and other such material may be created during the project. An example of this would be if, at the outset of your project, you were supplied with software by your supervisor that you incorporated in your final project you could not then 'market' the software that you produce in your project as your own. The University's stance on this is

---

*Queens do not lay any claim on IP generated by undergraduate students. The only exception is where the undergraduate has had input from a member of QUB staff and they have shared ownership rights in the invention - in such cases we will seek to protect the return to our member of staff.*

If you have any queries on this then contact me and I will put you in touch with the relevant body within Queen's that deals with IP.

## 18. Allocation of Marks

Problem Description, Solution approach and Work Plan (Semester 1)	10%
Interim Demonstration (Semester 1)	10%
Dissertation	35%
System Produced (including demonstration)	35%
Project Management	10%

## 19. What You are Required to Submit: A Summary

1. A **Problem Description** (max: 4 pages) should be emailed to your supervisor by Friday 28th October 2016, 4pm..
2. An **Updated Problem Description, Solution Approach and Work Plan** to contain the updated problem description, a solution approach including verifiable criteria for success of the project and a work plan (max: 8 pages) by Friday 9th December 2016 4pm .
3. An **Interim Demonstration**. This normally comprises a demonstration of a partially working system which satisfies a *substantial* number of the requirements agreed in advance with your supervisor. Demonstrations must be complete by Wednesday 14th December 2016 as arranged with your supervisor.
4. TWO copies of your bound dissertation to the CS main office before Tuesday 9th May 2017 4 pm. (max. 45 pages).
5. An electronic submission of your dissertation via Turnitin (details of how to do that by email/qol) by Wednesday 10th May 2017 4pm.
6. The Demonstration Day to Supervisors and Assessors will be held on Thursday 11th May 2017. **All students are required to attend demonstration day.** Details of the schedule will appear closer to the time.
7. A submission is not considered valid unless the Dissertation is handed into the Reception in the Computer Science building and the project register is signed. Dissertations must not be left in a lecturer's pigeon-hole. Dissertations must not be left with another member of university staff.

## 20. Dissertation and System Assessment

Each project dissertation will be assessed by the supervisor and up to two assessors not associated with the supervision of the project.

## 21. Late Submissions

All pieces of work must be submitted before the stated deadline.

**University regulation: a 5% penalty is incurred for each day that a submission is late.**

## **22. Hardware/Software Platform to be Used**

The projects must be developed on the hardware/software platform agreed with the project supervisor. You are responsible for ensuring that adequate precautions are made to prevent the loss of data.

## **23. Dissertation Submission Format**

**Two** spiral bound copies of your dissertation. Your name and the course title should be written on the **front cover**.

### **Document Layout – Documents not conforming to this will be returned.**

Documents must:

- be prepared using a word processor.
- Use Times Roman 12pt font.
- Have a page number on each page
- Have a Justified Layout with 2.5 cm (one inch) margin on all sides
- Must have a line spacing: 1.5 lines

Please read your submissions carefully and check document spelling using spell checking tools. The main text **should not** exceed 45 pages.

**Additionally, you are required to submit a pdf version via the Turnitin web service. More details of how to do this will follow. This is to check for plagiarism. Be aware that this service detects and highlights any text that is similar to existing sources.**

## **24. Background Reading**

Supervisors may provide references to suitable background material for the project. However, it is the student's responsibility to read around the area using books, articles and web based material. There are various search tools available via the library site (Use the Article search to obtain research papers). Google Scholar is also very useful.

## **25. Use of other resources**

- You must not make any use of any projects which are available online unless approved by your supervisor (any such projects should be referenced clearly). Project submissions will be checked for plagiarism (via the Turnitin).
- DO NOT copy text from other sources unless placed in quotes and cited.
- Even if you write something in your own words but it is based on an existing source, please place a citation in the document to the source.

## **26. Level of Project Difficulty**

The project is meant to stretch your ability. On the one hand, if you are given a problem which is beyond you, you may not be able to deliver anything much at all; on the other hand, if you are given a problem which is not challenging, it will not give you scope to do yourself justice, or the chance to gain high marks. You should try to realise your maximum potential.

While we will take your stated preferences into account in allocating projects, we cannot guarantee a project which meets all your preferences. Part of the training in this module is being able to take any task which is given to you, and complete it to the best of your ability.

**SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE**

**CSC3002 – COMPUTER SCIENCE PROJECT**

**Dissertation Cover Sheet**

A signed and completed cover sheet must accompany the submission of the Software Engineering dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

Student Name:

Student Number:

Project Title:

Supervisor:

**Declaration of Academic Integrity**

Before signing the declaration below please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

**I declare that I have read both the University and the School of Electronics, Electrical Engineering and Computer Science guidelines on plagiarism - <http://www.qub.ac.uk/schools/eeecs/Education/StudentStudyInformation/Plagiarism/> - and that the attached submission is my own original work. No part of it has been submitted for any other assignment and I have acknowledged in my notes and bibliography all written and electronic sources used.**

*Student's signature*

*Date of submission*