Faculty of Technology – Course work Specification 2017/18

Module name:	**Object Oriented Design and Development with C++**
Module code:	**IMAT2605**
Title of the Assignment:	**Re-engineering, securing and decommissioning software**

This coursework item is: (delete as appropriate)	Summative	
This summative coursework will be marked anonymously	No	

The learning outcomes that are assessed by this coursework are:

1. Be able to effectively communicate OO designs through UML static class diagrams and a set of documented C++ classes
5. Be able to maintain/re-engineer/decommission an existing piece of software improving design and security

This coursework is: (delete as appropriate)	Individual	

This coursework constitutes 50% **to the overall module mark.**

Date Set:	**4th October 2017**
Date & Time Due:	**11th of January 2018 4pm**

The 'normal' coursework return date for this work is:
8th of February 2018.
If for any reason this is not forthcoming by the due date your module leader will let you know why and when it can be expected. The Head of Studies (headofstudies-tec@dmu.ac.uk) should be informed of any issues relating to the return of marked coursework and feedback.

When completed you are required to submit your coursework to:

1. Report (Word or PDF document) to be submitted via a Turnitin link on blackboard.

Late submission of coursework policy: Late submissions will be processed in accordance with current University regulations which state:
*"the time period during which a student may submit a piece of work late without authorisation and have the work capped at 40% if passed is **14 calendar days**. Work submitted unauthorised more than 14 calendar days after the original submission date will receive a mark of 0%. These regulations apply to a student's first attempt at coursework. Work submitted late without authorisation which constitutes reassessment of a previously failed piece of coursework will always receive a mark of 0%."*

Academic Offences and Bad Academic Practices:
These include plagiarism, cheating, collusion, copying work and reuse of your own work, poor referencing or the passing off of somebody else's ideas as your own. If you are in any doubt about what constitutes an academic offence or bad academic practice you must check with your tutor. Further information is available at:
http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/academic-offences.aspx and
http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/bad-academic-practice.aspx

Tasks to be undertaken:

You have been given a piece of legacy software. You are to synthesise an object-oriented design for this software and document this design using static UML class

diagrams. The legacy software makes use of TCP/IP to allow the game to be played over a network. You must write a short report (no more than 6 sides of A4) describing the following:

- Choices/false starts/thinking around your final OO design.
- What the security risks are with this software and how you might mitigate these on an ongoing basis.
- Any issues which would need to be addressed when decommissioning the existing software and deploying your new design and how these might be overcome.

Deliverables to be submitted for assessment:

- A set of UML static class diagrams giving your OO redesign for the legacy software.
- A short report of no more than six sides of A4 containing the following:
    - Choices/false starts/thinking around your final OO design.
    - What the security risks are with this software and how you might mitigate these on an ongoing basis.
    - Any issues which would need to be addressed when decommissioning the existing software and deploying your new design and how these might be overcome.

How the work will be marked:

You mark will be based on how much of the software you managed to redesign and how well you redesigned it terms of problem decomposition, maintainability and to some extent elegance.

The mark will be assessed using the following mark scheme:

- 0 – 20%
  Little of the legacy code base has been redesigned and the other discussion is missing.
- 20 – 40%
  Little of the legacy code base has been redesigned and the other discussion is weak.
- 40 – 50%
  The majority of the client side legacy code base has been redesigned and the other discussion is weak.
- 50 – 60%
  Virtually the entire client side legacy code base has been redesigned and the other discussion is has some merit but is lacking in key areas.
- 60 – 70%
  Virtually the entire client side legacy code base has been redesigned and some attempt has been made to redesign the server side code to fit with the OO paradigm. The other discussion is covers the main points around security, cyber security management and decommissioning of software.
- 70 – 85%
  Virtually the entire client and server side legacy code base has been redesigned and some attempt has been made to redesign the server side code to fit with the OO paradigm. The other discussion is covers the main points around security, cyber security management and decommissioning of software.

- 85 – 100%
  All of client and server side legacy code base has been redesigned in line with the OO paradigm. A thorough and rigorous discussion is presented which covers security, cyber security management and decommissioning of software.

It should be clear that these criteria are subjective and open to interpretation. To ensure a high mark you need to achieve a good design with clear discussion of security, cyber security management and software decommissioning.
Your report will be run through plagiarism checking software. If you have used material from any source you must reference this source, failure do so will lead to you receiving a mark of zero and may be referred to the academic offences officer. For more information on referencing see Referencing and Citation in the CLASS HEAT document at http://libguides.library.dmu.ac.uk/class/HEAT

Module leader/tutor name:	**Simon Coupland**
Contact details:	**simonc@dmu.ac.uk**