1. **Title of the module**

COMP8860 (CO886) – Software Engineering

1. **School or partner institution which will be responsible for management of the module**

School of Computing

1. **The level of the module (Level 4, Level 5, Level 6 or Level 7)**

Level 7

1. **The number of credits and the ECTS value which the module represents**

15 credits (7.5 ECTS)

1. **Which term(s) the module is to be taught in (or other teaching pattern)**

Spring

1. **Prerequisite and co-requisite modules**

COMP8820 Advanced Object-Oriented Programming, or COMP8710 Advanced Java for Programmers

1. **The programmes of study to which the module contributes**

Portfolio of Taught Postgraduate Programmes in Computing

1. **The intended subject specific learning outcomes.  
   On successfully completing the module students will be able to demonstrate:**
   1. A critical understanding of the principles and practice of the development of software systems (broadly defined) including requirements specification, design, validation, implementation and evolution.
   2. A comprehensive understanding of techniques for modelling software systems and their domains.
   3. The ability to design and implement test plans, and apply a wide variety of testing techniques effectively and efficiently, and being able to evaluate their efficacy in identifying a wide range of faults.
   4. The conceptual understanding of planning, documentation, estimation, quality, time, cost and risk evaluation in the business context.
   5. Self-direction in the design of software systems, including design simplicity, appropriateness and styles of system thinking and focused problem solving.
   6. Critical awareness of the deployment of certain types of software system and show an understanding of the professional duties of software, their societal responsibilities as well as critical understanding of fundamental limitations of different kinds of software systems.
   7. Ability to use state-of-the-art tools and techniques when developing software systems.
2. **The intended generic learning outcomes.  
   On successfully completing the module students will be able to:**
   1. The ability to work effectively as a member of a team.
   2. Effective use of IT facilities.
   3. Time management and organisational skills, including the ability to manage one’s own learning and development.
   4. An understanding of the importance of keeping quality procedures under review, to ensure that they fulfil quality objectives cost-effectively, and in a manner understood by the project team.
   5. Be able to analyse a problem specification and to design and implement a solution
   6. Appreciation of the social, ethical and professional issues related to software development.
   7. Appreciation of the importance of continued professional development as part of lifelong learning.
3. **A synopsis of the curriculum**

* Software processes.
* Modelling techniques, and the use of these techniques throughout the project lifecycle.
* Introduction to modelling principles (decomposition, abstraction, generalization, projection/views) and types of models (information, behavioural, structural, domain and functional).
* Risk and risk management in software.
* Approaches to software testing and inspection.
* Approaches to software configuration management.
* Software engineering tools: configuration control, project management, integrated development environments and modelling tools.

1. **Reading list (Indicative list, current at time of publication. Reading lists will be published annually)**

Cohn, M. (2002). *User Stories Applied*. Addison-Wesley.

Fowler, M (2003). *UML Distilled: A Brief Guide to Standard Object Modeling Language*. Addison-Wesley.

Sommerville, I. (2015). *Software Engineering* 10th ed. Addison-Wesley.

Craig, RD. and Jaskie, SP. (2002). *Systematic Software Testing*. Artech House.

Pezze, M, Young, M. (2007). *Software Testing and Analysis: Process, Principles and Techniques*. John Wiley & Sons.

McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press.

Hall, EM. (1998). *Managing Risk: Methods for Software Systems Development*. Addison-Wesley.

Martin, R. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.

Brooks, F.P. (1995). *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley Professional.

Ensmenger, N. L. (2010). *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press.

Slayton, R. (2013). *Arguments that Count: Physics, Computing, and Missile Defense, 1949-2012*. The MIT Press.

1. **Learning and teaching methods**

Total contact hours: 32 hours

Private study hours: 118 hours

Total study hours: 150 hours

1. **Assessment methods**
   1. Main assessment methods

Case Studies, 1000 words (15%)

Group Project, undertaken and assessed on group basis, 40 hours (35%)

Examination, 2 hours (50%)

13.2 Reassessment methods

Like for like

1. ***Map of module learning outcomes (sections 8 & 9) to learning and teaching methods (section12) and methods of assessment (section 13)***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Module learning outcome** | 8.1 | 8.2 | 8.3 | 8.4 | 8.5 | 8.6 | 8.7 | 9.1 | 9.2 | 9.3 | 9.4 | 9.5 | 9.6 | 9.7 |
| **Learning/ teaching method** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Private study | X | X | X | X | X | X | X |  | X | X |  |  |  |  |
| Lectures | X | X | X | X | X | X |  |  |  | X | X | X | X | X |
| Practical classes |  | X |  |  |  |  | X | X | X |  |  |  |  |  |
| Group project |  | X |  |  |  |  | X | X | X | X |  |  |  |  |
| **Assessment method** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Examination | X | X | X | X | X | X |  |  |  |  |  | X |  |  |
| Group project | X | X | X | X | X | X | X | X | X | X | X | X |  |  |
| Case studies |  |  |  |  |  | X |  | X |  | X |  |  | X | X |

1. **Inclusive module design**

The School recognises and has embedded the expectations of current equality legislation, by ensuring that the module is as accessible as possible by design. Additional alternative arrangements for students with Inclusive Learning Plans (ILPs)/declared disabilities will be made on an individual basis, in consultation with the relevant policies and support services.

The inclusive practices in the guidance (see Annex B Appendix A) have been considered in order to support all students in the following areas:

a) Accessible resources and curriculum

b) Learning, teaching and assessment methods

1. **Campus(es) or centre(s) where module will be delivered**

Canterbury

1. **Internationalisation**

The topics addressed by this module relate to a field which is of international importance, given the global role of computers in today's technological innovation. The topics covered by this module are international in nature, being identical worldwide and independent of traditional spoken language.

**DIVISIONAL USE ONLY**

**Revision record – all revisions must be recorded in the grid and full details of the change retained in the appropriate committee records.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date approved | Major/minor revision | Start date of the delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
| 04/12/2020 | Minor | September 2021 | 8,11,13 | No |
|  |  |  |  |  |