1. **Title of the module**

COMP8940 (CO894) - Development Frameworks

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)**

Autumn

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

Co-requisite: COMP8710: Advanced Java for Programmers or equivalent experience

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

Portfolio of Taught Postgraduate Programmes in Computing

(compulsory for MSc Advanced Software Development).

1. **The intended subject specific learning outcomes.
On successfully completing the module students will be able to:**

8.1 be able to make effective use of software development environments and frameworks for developing, debugging, testing and deploying applications;

8.2 have an understanding of the requirements and facilities of industry-standard software development, testing and deployment environments;

8.3 have an understanding of how development environments integrate into the project life-cycle, including making effective use of project and source-code management tools;

8.4 have an understanding of the conceptual basis and evolution of software frameworks and their relationship with software development environments.

1. **The intended generic learning outcomes.
On successfully completing the module students will be able to:**

9.1 identify, analyse and formulate criteria and specifications appropriate to a given problem;

9.2 manage their own time effectively, in the completion of coursework and private study;

9.3 work effectively as part of a team;

9.4 critically evaluate commercially-relevant software products.

1. **A synopsis of the curriculum**

Introduction to software development environments and the facilities they provide. Development of simple applications in these environments, using a broad range of the facilities provided. Software libraries and frameworks, and their use in developing and testing software systems. Use of development frameworks’ facilities for project and source-code management, automated testing, refactoring and profiling. Deploying applications across multiple platforms using installers and build-systems, continuous integration and deployment.

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

Jim D'Anjou, Scott Fairbrother, Dan Kehn, John Kellerman, Pat McCarthy, "The Java

Developer's Guide to Eclipse", Addison Wesley; 2nd ed., 2004. ISBN: 978-0321305022.

C. Pilato, Ben Collins-Sussman, Brian Fitzpatrick, "Version Control with Subversion",

O'Reilly; 2nd ed., 2008. ISBN: 978-0596510336.

Peter Smith, “Software Build Systems: Principles and Experience”, Addison Wesley, 2011,

ISBN: 978-0321717283

Jez Humble, David Farley, “Continuous Delivery: Reliable Software Releases through Build,

Test, and Deployment Automation”, Addison Wesley, 201, ISBN: 978-0321601919.

1. **Learning and teaching methods**

Total contact hours: 22

Private study hours: 128

Total study hours: 150

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

Two reports on features and use of categories of frameworks, 1000 words each (33%)

Group Presentation, 10 hours (17%)

Exam, 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* | *9.1* | *9.2* | *9.3* | *9.4* |  |  |
| **Learning/ teaching method** |  |  |  |  |  |  |  |  |  |  |
| Lectures | X | X | X | X | X |  |  | X |  |  |
| Private Study | X | X | X | X | X | X | X | X |  |  |
| **Assessment method** |  |  |  |  |  |  |  |  |  |  |
| Reports | X | X | X | X | X | X |  | X |  |  |
| Presentation | X | X | X |  | X | X | X | X |  |  |
| Exam |  | 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.

In addition, this module explicitly addresses issues in the creation of software for use in multiple locales.

**FACULTIES SUPPORT OFFICE 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) |
|  |  |  |  |  |
|  |  |  |  |  |

Revised FSO Jan 2018