1. **Title of the module**

COMP8820 (CO882) - Advanced Object-Oriented Programming

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**

Pre-requisite: COMP8820: Object-Oriented Programming

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:**

8.1 develop non-trivial computer programs following recognized object-oriented principles.

8.2 critically evaluate the suitability of a commercially-relevant implementation language in the solution of particular problems.

8.3 describe concepts used in programming and to discuss programming using vocabulary from professional computer science.

8.4 choose and use appropriate data structures and algorithms in the construction of programs.

8.5 apply principled design techniques in the construction of software.

8.6 choose and use appropriate software testing strategies.

8.7 critically reflect on both the process and outcomes of software creation.

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

9.1 communicate with other professionals using appropriate technical vocabulary.

9.2 critically reflect on and evaluate professional practice.

9.3 write about technical material in an appropriate voice.

9.4 discover and use professional and technical documentation.

1. **A synopsis of the curriculum**

Building upon Introduction to Object-Oriented Programming, this module covers the design and implementation of high-quality software using OO techniques. Systems are modelled as configurations of objects communicating with one another. Techniques (e.g. inheritance) are introduced which allow objects to play different roles within a system. These two concepts are key to the support for adaptation and reuse that OOP provides. Much emphasis will be placed on gaining a deep understanding of these concepts and applying them in practice by developing programs in Java. The remainder of the module will explore software component frameworks, specifically those that come packaged with Java, placing most emphasis on the frameworks to support the structuring and manipulation of data (data structures and algorithms).

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

Barnes, D.J. and Kölling, M. (2017) Objects First with Java - A Practical Approach using BlueJ (6th Edition): Pearson Education

1. **Learning and teaching methods**

Total contact hours: 25

Private study hours: 125

Total study hours: 150

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

Written assessment (40%)

Two programming assessments with reports (60% total)

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* |
| **Learning/ teaching method** |  |  |  |  |  |  |  |  |  |  |  |  |
| Lectures | x | x | x | x | x | x |  |  |  |  |  |  |
| Classes | x |  |  | x | x | x |  |  | x | x | x | x |
| Private study | x | x | x | x | x | x | x |  | x | x | x | x |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Assessment method** |  |  |  |  |  |  |  |  |  |  |  |  |
| Written assessment |  | x | x | x |  |  | x |  | x | x | x | x |
| Programming assessments with reports | x | x | x | x | x | x | 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.

**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