1. **Title of the module**

COMP3830 (CO383) Problem Solving with Algorithms

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

Division of Computing, Engineering, Mathematical Sciences (CEMS)

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

Level 4

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

15

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

Autumn or Spring

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

Pre-requisite: COMP3200 (CO320): Introduction to Object-Oriented programming

1. **The course(s) of study to which the module contributes**

BSc Computer Science, including all variants, both with and without Year in Industry.

BSc Business Information Technology, BSc Computing, both with and without Year in Industry.

BSc Artificial Intelligence, BSc Data Science, BSc Software Engineering, both with and without Year in Industry.

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

8.1 Read a problem description and apply an appropriate algorithm to solve that problem.

8.2 Formulate a solution to a problem in algorithmic form using pseudocode.

8.3 Reason about the correctness of an algorithm.

8.4 Reason about the runtime of an algorithm.

8.5 Implement an algorithm as a part of an executable program.

8.6 Implement basic data structures (e.g., arrays, lists, trees) and use algorithmic techniques (recursion and divide & conquer) to solve well-known problems (searching and sorting), but also how to apply their principles to newly encountered problems.

1. **The intended generic learning outcomes.  
   On successfully completing the module students will be able to:**
   1. Work in teams.
   2. Communicate their understanding of technical problems and their solutions.
   3. Make effective use of IT facilities.

9.4 Manage their time and resources effectively.

1. **A synopsis of the curriculum**

This module aims to strengthen the foundational programming-in-the-small abilities of students via a strong, practical, problem solving focus. Specific topics will include introductory algorithms, algorithm correctness, algorithm runtime, as well as big-O notation. Essential data structures and algorithmic programming skills will be covered, such as arrays, lists and trees, searching and sorting, recursion, and divide and conquer.

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

Skiena, Steven S, “The Algorithm Design Manual”, Springer, 2008.

1. **Learning and teaching methods**

Total contact hours: 32

Private study hours: 118

Total study hours: 150

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

50%: two in-course tests of two hours each. One worth 20% and the other 30%

10%: programming assessment, approximately 15 hours of work.

10%: programming assessment, approximately 15 hours of work.

10%: group programming assessment, approximately 15 hours of work.

10%: programming assessment, approximately 15 hours of work.

10%: programming assessment, approximately 15 hours of work.

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 | 9.1 | 9.2 | 9.3 | 9.4 |
| **earning/ teaching method** |  |  |  |  |  |  |  |  |  |  |
| **Private Study** | **X** | **X** | **X** | **X** | **X** | **X** |  |  | **X** | **X** |
| Classes | **X** |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
| Lectures | **X** | **X** | **X** | **X** |  |  |  |  |  |  |
| **Assessment method** |  |  |  |  |  |  |  |  |  |  |
| In-course tests | **X** | **X** | **X** | **X** |  |  |  |  | **X** |  |
| Programming assessments | **X** |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |

1. **Inclusive module design**

The Division 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 delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
| 10/11/2020 | Minor |  | 7 | No |
|  |  |  |  |  |