1. **Title of the module**

COMP5570 (CO557) Computer Systems

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 5

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

15 (7.5 ECTS)

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 modules:

* CO520 (Further Object-Oriented Programming),
* CO3BB (Problem Solving with Algorithms),
* CO3AA (Computers and the Cloud)
1. **The programmes of study to which the module contributes**

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

1. **The intended subject specific learning outcomes.
On successfully completing the module students will be able to:**
2. Describe the purpose of, and the interaction between, the hardware and software components of a typical computer system.
3. Appreciate the principles of abstraction and layering in building today’s complex systems and be able to apply them.
4. Identify the interfaces of abstraction layers and be able to select an appropriate layer on which to build useful systems.
5. Describe comprehensively how programs are executed by each of the abstractions and critically assess the consequences of the abstractions on how we build systems today.
6. **The intended generic learning outcomes.
On successfully completing the module students will be able to:**
7. Communicate their understanding of technical problems and their solutions.
8. Make effective use of IT facilities.
9. Manage their time and resources effectively.
10. **A synopsis of the curriculum**

This module aims to provide students with an understanding of the fundamental components (hardware and software) of a typical computer system, and how they collaborate to execute software programs. The module provides a compressive overview from the lowest level of abstractions in hardware to the highest level of abstractions of modern programming languages. For example, they will see logic circuits, machine language, programming language implementations, high-level languages, and applications. This material provides a general understanding of computers, and it will also prepare students to develop software considering the system perspective, e.g. cost of abstraction and performance implications.

1. **Reading list (Indicative list, current at time of publication. Reading lists will be published annually)**
* Noam Nisan, Shimon Schocken (2005) The Elements of Computing Systems: Building a Modern Computer from First Principles. MIT Press, 344 pp.

Any additional textbooks for background reading will be advised by individual lecturers.

1. **Learning and teaching methods**

Total contact hours: 22 hrs

Total private study hours: 128 hrs

Total module study hours: 150 hrs

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

Project 1 (approximately 15 hours): 12%

Project 2 (approximately 15 hours): 12%

Project 3 (approximately 15 hours): 12%

Project 4 (approximately 15 hours): 14%

Examination (2 hrs) – 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 |
| **Learning/ teaching method** |  |  |  |  |  |  |  |
| Private Study | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| Lectures | **X** | **X** | **X** | **X** | **X** |  |  |
| **Assessment method** |  |  |  |  |  |  |  |
| Project 1 |  | **X** | **X** | **X** | **X** | **X** | **X** |
| Project 2 |  | **X** | **X** | **X** | **X** | **X** | **X** |
| Project 3 |  | **X** | **X** | **X** | **X** | **X** | **X** |
| Project 4 |  | **X** | **X** | **X** | **X** | **X** | **X** |
| Examination | **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 delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
|  |  |  |  |  |
|  |  |  |  |  |