

**This Version is No Longer Current**  
The latest version of this module is available [here](#)

## MODULE DESCRIPTOR

### Module Title

Introduction to Computing

|           |             |             |        |
|-----------|-------------|-------------|--------|
| Reference | CM1100      | Version     | 2      |
| Created   | April 2017  | SCQF Level  | SCQF 7 |
| Approved  | July 2016   | SCQF Points | 30     |
| Amended   | August 2017 | ECTS Points | 15     |

### Aims of Module

To provide students with a context for the computing discipline and an introduction to the skills needed to design, develop and evaluate solutions to simple programming problems.

### Learning Outcomes for Module

On completion of this module, students are expected to be able to:

- 1 Identify and demonstrate practical knowledge of the basic programming structures of sequence, selection and iteration to implement solutions to a computational problem.
- 2 Select and apply effective algorithms and simple data structures in the solution of a programming exercise.
- 3 Select and implement a systematic and structured approach to the design, development and evaluation of solutions to problems in the computing domain.
- 4 Describe the main components of a computer system, how software executes on a computer and have an introduction to associated legal, social, ethical, professional and security issues.
- 5 Select and use appropriate tools to develop simple models of software solutions.

### Indicative Module Content

The computer: Main components, development of hardware and software, LSEPIs and security issues. Software Development: Variable, data types, declarations and expressions, iterative and conditional programming constructs, functions/methods, parameter passing mechanisms and arrays. Algorithms: Introduction to classic algorithms including max, min, searching, sorting. Introduction to Computer Graphics: Pixels, colour, simple image processing algorithms. Modelling and Design: The software lifecycle, data flow diagrams, interaction diagrams, use case, documentation, stepwise refinement. Testing: Tracing programs. Strategies for testing, use of defensive programming. Standards and Best Practice Guides: ISO 20000, ISO 27001.

### Module Delivery

Key concepts on design and development practices are introduced through the lectures. The main emphasis of the course will be focused on the lab sessions. The lab sessions will create a flexible teaching session where individual lab assignments and group development will be interspersed with demonstrations of current techniques and practices. This combination will allow students to develop an understanding of the theoretical underpinning of modern programming structures, whilst promoting development of proficiency in the practical application of software development. Exposing the students to real world applications, demos, and case studies will help nurture entrepreneurial skills and ensure the continued relevance of the learning experience.

### Indicative Student Workload

|   | Full Time | Part Time |
|---|-----------|-----------|
| Contact Hours   | 84        | N/A       |
| Non-Contact Hours   | 216       | N/A       |
| Placement/Work-Based Learning Experience [Notional] Hours             | N/A       | N/A       |
| TOTAL   | 300       | N/A       |
| Actual Placement hours for professional, statutory or regulatory body |           |           |

### ASSESSMENT PLAN

If a major/minor model is used and box is ticked, % weightings below are indicative only.

#### Component 1

|              |                            |            |     |                    |      |
|--------------|----------------------------|------------|-----|--------------------|------|
| Type:        | Practical Exam             | Weighting: | 30% | Outcomes Assessed: | 4, 5 |
| Description: | A closed book examination. |            |     |                    |      |

#### Component 2

|              |                   |            |     |                    |         |
|--------------|-------------------|------------|-----|--------------------|---------|
| Type:        | Coursework        | Weighting: | 70% | Outcomes Assessed: | 1, 2, 3 |
| Description: | A practical exam. |            |     |                    |         |

### MODULE PERFORMANCE DESCRIPTOR

#### Explanatory Text

The calculation of the overall grade for this module is based on 30% weighting of C1 and 70% weighting of C2. An overall minimum grade D is required to pass the module.

|                 |   | Examination:   |   |   |   |   |   |    |
|-----------------|---|--|---|---|---|---|---|----|
|                 |   | A  | B | C | D | E | F | NS |
| Practical Exam: | A | A  | A | B | B | C | E |    |
|                 | B | B  | B | B | C | D | E |    |
|                 | C | B  | C | C | C | D | E |    |
|                 | D | C  | C | D | D | D | E |    |
|                 | E | D  | D | D | E | E | E |    |
|                 | F | E  | E | E | E | F | F |    |
| NS              |   | Non-submission of work by published deadline or non-attendance for examination |   |   |   |   |   |    |

**Module Requirements**

|                          |   |
|--------------------------|---|
| Prerequisites for Module | None, in addition to course entry requirements. |
| Corequisites for module  | None.   |
| Precluded Modules        | None.   |

**INDICATIVE BIBLIOGRAPHY**

- 1 HAVERBEKE, M., 2011. Eloquent JavaScript: A Modern Introduction to Programming. No Starch Press. 978-1-593-27282-1.
- 2 DIONISIO, J.D. and TOAL, R., 2011. Programming With Javascript: Algorithms And Applications For Desktop And Mobile Browsers. Jones and Bartlett Learning. 978-0-763-78060-9.
- 3 MCLAUGHLIN, B., POLLICE, G. and WEST, D., 2009. Head First Object-Oriented Analysis and Design. Safari Books Online. 978-0-596-55675-4.
- 4 SPRAUL, V.A., 2012. Think Like a Programmer: An Introduction to Creative Problem Solving. No Starch Press. 1593274246.
- 5 HORSTMANN, C., 2013. Big Java: Late Objects. 1st ed. John Wiley.