

This Version is No Longer Current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

MODULE DESCRIPTOR

Module Title

Introduction to Programming

| | | | |
|-----------|-------------|-------------|--------|
| Reference | CM1112 | Version | 3 |
| Created | July 2023 | SCQF Level | SCQF 7 |
| Approved | March 2021 | SCQF Points | 15 |
| Amended | August 2023 | ECTS Points | 7.5 |

Aims of Module

To provide students with the competencies needed to design and develop simple computer programs.

Learning Outcomes for Module

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

- 1 Identify basic programming structures of sequence, selection and iteration when implementing solutions to a computational problem.
- 2 Implement a structured approach to the design, development and testing of solutions to programming problems.
- 3 Select appropriate algorithms and simple data structures for use in the solution of programming exercises.
- 4 Employ appropriate software tools to create effective solutions to programming problems.

Indicative Module Content

Software Development: Variable, data types, declarations and expressions, iterative and conditional programming constructs, functions/methods, parameter passing mechanisms and arrays. Objects. Simple algorithms including max, min. Strategies for testing. Simple applications of programming to the solution of computational problems

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 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.

Indicative Student Workload

| | Full Time | Part Time |
|--|-----------|-----------|
| Contact Hours | 40 | N/A |
| Non-Contact Hours | 110 | N/A |
| Placement/Work-Based Learning Experience [Notional] Hours | N/A | N/A |
| TOTAL | 150 | N/A |
| <i>Actual Placement hours for professional, statutory or regulatory body</i> | | |

ASSESSMENT PLAN

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

Component 1

| | | | | | |
|--------------|--|------------|------|--------------------|------------|
| Type: | Coursework | Weighting: | 100% | Outcomes Assessed: | 1, 2, 3, 4 |
| Description: | Group coursework covering all learning outcomes. | | | | |

MODULE PERFORMANCE DESCRIPTOR**Explanatory Text**

This module is assessed on a pass/unsuccessful basis. The Module Grade is based on performance in Component 1 (coursework) as detailed below.

| Module Grade | Minimum Requirements to achieve Module Grade: |
|--------------|--|
| Pass | Pass in C1. |
| Fail | Fail, i.e. unsuccessful, in C1. |
| 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., 2018. Eloquent JavaScript: A Modern Introduction to Programming, 3rd ed. No Starch Press. ISBN-13: 978-1593279509. <https://eloquentjavascript.net/>
- 2 DAVIS, A., 2020. Modern Programming Made Easy. Apress, Berkeley, CA. ISBN-13: 978-1-4842-5568-1.
- 3 THOMAS, D. and HUNT, A., 2019. The Pragmatic Programmer: Your journey to mastery, 20th Anniversary Edition. Addison Wesley. ISBN-13: 978-0135957059
- 4 SPRAUL, V.A., 2012. Think Like a Programmer: An Introduction to Creative Problem Solving. No Starch Press. ISBN-13: 978-1593274245
- 5 VICKERS, P., 2008. How to Think like a Programmer: Problem Solving for the Bewildered. Cengage Learning EMEA. ISBN-13: 978-1408065822