STUDENT AND ACADEMIC SERVICES

# MODULE SPECIFICATION

| Part 1: Information | | | |
|---|---|---|---|
| Module Title | Programming | | |
| Module Code | UFCFEU-30-1 | Level | Level 4 |
| For implementation from | 2020-21 | | |
| UWE Credit Rating | 30 | ECTS Credit Rating | 15 |
| Faculty | Faculty of Environment & Technology | Field | |
| Department | FET Dept of Computer Sci & Creative Tech | | |
| Module type: | Standard | | |
| Pre-requisites | None | | |
| Excluded Combinations | None | | |
| Co- requisites | None | | |
| Module Entry requirements | None | | |

## Part 2: Description

**Overview**: It is the intent of this module to teach students to basics of programming. It introduces students to the core concepts of programming with an introduction to algorithms and the characteristics of programming paradigms. Appropriate programming languages will be chosen to illustrate the concepts.

**Educational Aims:** Contributes to underpinning technical knowledge.

**Outline Syllabus:** • algorithms and program design

- fundamental programming concepts
- fundamental data structures
- typical program development environment and methods
- object-oriented programming
- functional programming
- event driven and reactive programming
- language translation and execution
- syntax analysis
- compiler semantic analysis;
- code generation

- coding in assembly language
- machine code
- scripting languages
- database query language
- the different aspects of the software development lifecycle and how they combine to deliver successful outcome, for example:

need, design, trade-offs, implementation, deployment, support, evolution, validation, verification and assurance

- different approaches to developing software, including sequential, iterative/agile, etc.
- advantages and disadvantages of different software development processes along with choice of process in different contexts.
- selection and use of different tools and environments that support software development at different stages in the lifecycle
- the principles of systems engineering, including all aspects of technology, people, culture and process and the environment within which a system of interest exists and operates
- the benefits of a system approach to dealing with challenges arising from complexity, emergence, adaption and co-evolution

**Teaching and Learning Methods:** Lecture sessions cover the technical knowledge required. Designated practical work is included to ensure that students have absorbed and understood the key principles involved.

## Part 3: Assessment

This module is assessed by two methods: practical coursework and an examination (2 hours)

Component A: Unseen examination (2 hours)
Apprentices will be assessed on their knowledge and application of development lifecycles, methodologies and processes.

Component B: Portfolio of coursework (fully documented)
Students will demonstrate their grasp of core programming concepts through a number of practical exercises of increasing complexity and difficulty. For example, they will design algorithms and implement a simple program for three different scenarios, storing and retrieving data in a suitable system.
In each case there must be a technical description, code and evidence of testing and correct function. A short report will explain the development methodologies used in each case.

An unseen exam will be used to assess the students knowledge of overarching development life-cycles, methodologies and processes.

| First Sit Components | **Final Assessment** | **Element weighting** | **Description** |
|---|---|---|---|
| Examination - Component A | | 40 % | 2 hour exam to test process knowledge. |
| Portfolio - Component B | ✓ | 60 % | A fully documented portfolio of practical exercises. |
| Resit Components | **Final Assessment** | **Element weighting** | **Description** |
| Examination - Component A | | 40 % | 2 hour exam |
| Portfolio - Component B | ✓ | 60 % | Re-worked fully documented portfolio of practical exercises. |

## Part 4: Teaching and Learning Methods

| | |
|---|---|
| Learning Outcomes | On successful completion of this module students will achieve the following learning outcomes: |

| Module Learning Outcomes | Reference |
|---|---|
| Write, test, and debug programs in high and low level languages and scripts. | MO1 |
| Apply system engineering and software development methodologies and models. | MO2 |
| Design and implement algorithms in different languages within a suitable Integrated Development Environment (IDE). | MO3 |
| Explain and compare development life-cycles and processes. | MO4 |

Contact Hours

| | |
|---|---|
| **Independent Study Hours:** | |
| Independent study/self-guided study | 135 |
| **Total Independent Study Hours:** | 135 |
| **Placement Study Hours:** | |
| Placement | 75 |
| **Total Placement Study Hours:** | 75 |
| **Scheduled Learning and Teaching Hours:** | |
| Face-to-face learning | 90 |
| **Total Scheduled Learning and Teaching Hours:** | 90 |
| **Hours to be allocated** | 300 |
| **Allocated Hours** | 300 |

| | |
|---|---|
| Reading List | *The reading list for this module can be accessed via the following link:*<br>https://rl.talis.com/3/uwe/lists/6CFAF645-C8F0-4DE0-2AD8-F7B85AB7F809.html |

**Part 5: Contributes Towards**

This module contributes towards the following programmes of study:

BSc (Hons) Cyber Security Technical Professional (integrated degree) BSc (Hons) 2020-21