



## Module Definition Form (MDF)

|                               |   |
|-------------------------------|---|
| <b>Module code: MOD003484</b> | <b>Version: 6 Date Amended: 27/Nov/2025</b> |
|-------------------------------|---|

|                        |
|------------------------|
| <b>1. Module Title</b> |
| Software Principles    |

|                          |
|--------------------------|
| <b>2a. Module Leader</b> |
| Md Hasan                 |

|  |
|--|
| <b>2b. School</b>                            |
| School of Computing and Information Sciences |

|                                    |
|------------------------------------|
| <b>2c. Faculty</b>                 |
| Faculty of Science and Engineering |

|                  |
|------------------|
| <b>3a. Level</b> |
| 4                |

|                        |
|------------------------|
| <b>3b. Module Type</b> |
| Standard (fine graded) |

|                    |
|--------------------|
| <b>4a. Credits</b> |
| 30                 |

|                        |
|------------------------|
| <b>4b. Study Hours</b> |
| 300                    |

| <b>5. Restrictions</b>                             |             |             |           |
|--|-------------|-------------|-----------|
| Type   | Module Code | Module Name | Condition |
| Pre-requisites:                                    | None        |             |           |
| Co-requisites:                                     | None        |             |           |
| Exclusions:  | None        |             |           |
| <b>Courses to which this module is restricted:</b> |             |             |           |

## LEARNING, TEACHING AND ASSESSMENT INFORMATION

### 6a. Module Description

In this module, you'll be introduced to the fundamental concepts needed to understand, design, implement, and test programs written in high-level programming languages using a range of fundamental data types, control structures, and basic data structures. You'll learn a structured design methodology that supports the development of clear, succinct, and logical programs, from the initial idea through to design, implementation, and testing. You will also explore the features provided by modern interactive development environments (IDEs) that support the programmer.

By the end of the module, you'll be able to analyse a simple set of written requirements and design, implement, and test code that meets those requirements using appropriate variables, control and data structures using correct language syntax, and know how to approach program testing and debugging. You will also demonstrate familiarity with the taxonomy of programming languages and the software development life cycle.

### 6b. Outline Content

Explore and experiment using taught methodological techniques to stimulate, explore, develop and communicate ideas  
Background to the nature of Programming languages  
Software Development life-cycle  
Employ basic management techniques to facilitate communication and record keeping in a team situation  
Investigate and use established design methodologies  
Explore the nature of algorithms  
Use basic numerical and logical operators and functions to manipulate simple and multi-dimensional data structures  
Use iteration and selection techniques to navigate simple and multi-dimensional data structures  
Use a range of graphical user interface techniques to interact with users of software

### 6c. Key Texts/Literature

The reading list to support this module is available at: <https://readinglists.aru.ac.uk/>

### 6d. Specialist Learning Resources

Aptana Studio 3 or other IDE

| 7. Learning Outcomes (threshold standards) |   |  |
|--|---|--|
| No.  | Type  | On successful completion of this module the student will be expected to be able to:  |
| 1  | Knowledge and Understanding                                 | Use a range of techniques for problem decomposition, decision modelling and algorithm design such that, within a simple problem description, appropriate variables, decisions and repetitive actions can be identified and translated into appropriate code constructs |
| 2  | Knowledge and Understanding                                 | To understand programming language families, and competing models of the software development life-cycle   |
| 3  | Intellectual, practical, affective and transferrable skills | To select appropriate tools and techniques to analyse and design solutions to specific simple scenarios  |
| 4  | Intellectual, practical, affective and transferrable skills | To create simple scripts to perform a number of operations in turn to achieve a desired effect and to understand the purpose and syntax required to perform conditional and loop statements  |
| 5  | Intellectual, practical, affective and transferrable skills | To work effectively as a team player, understand different roles within a team and how to administrate effectively within that role  |

| 8a. Module Occurrence to which this MDF Refers |            |   |          |                  |
|--|------------|---|----------|------------------|
| Year   | Occurrence | Period                                      | Location | Mode of Delivery |
| 2025/6   | ZZF        | Template For Face To Face Learning Delivery |          | Face to Face     |

| 8b. Learning Activities for the above Module Occurrence |       |                   |   |
|---|-------|-------------------|---|
| Learning Activities                                     | Hours | Learning Outcomes | Details of Duration, frequency and other comments   |
| Lectures  | 36    | 1 - 5             | 3 hrs / wk lecture  |
| Other teacher managed learning                          | 24    | 1 - 5             | 2 hrs / wk lab  |
| Student managed learning                                | 240   | 1 - 5             | Practicing skills and techniques learnt to master them in order to complete assessment requirements and to be able to understand the following lectures. Some research will be required in order to fully complete the assignments. |
| TOTAL:  | 300   |                   |   |

| 9. Assessment for the above Module Occurrence |                   |                   |               |                         |                     |
|---|-------------------|-------------------|---------------|-------------------------|---------------------|
| Assessment No.                                | Assessment Method | Learning Outcomes | Weighting (%) | Fine Grade or Pass/Fail | Qualifying Mark (%) |
| 010   | Coursework        | 1-5               | 100 (%)       | Fine Grade              | 30 (%)              |
| <b>Coursework</b>                             |                   |                   |               |                         |                     |

| Assessment components for Element 010 |   |                   |               |   |
|---------------------------------------|---|-------------------|---------------|---|
| Component No.                         | Assessment Title  | Submission Method | Weighting (%) | Components needed for Mark Calculation? |
| 010/1                                 | Coursework Project Proposal – 1,500 Words                 | Canvas            | 30 (%)        | All                                     |
| 010/2                                 | Design and implementation report equivalent to 3500 words | Canvas            | 70 (%)        |   |

In order to pass this module, students are required to achieve an overall mark of 40% (for modules at levels 3, 4, 5 and 6) or 50% (for modules at level 7\*).

In addition, students are required to:

(a) achieve the qualifying mark for each element of fine graded assessment as specified above

(b) pass any pass/fail elements

[\* the pass mark of 50% applies for all module occurrences from the academic year 2024/25 – see Section 3a of this MDF to check the level of the module and Section 8a of this MDF to check the academic year]