



## Module Definition Form (MDF)

<b>Module code: MOD003212</b>	<b>Version: 11 Date Amended: 27/Nov/2025</b>
-------------------------------	--

<b>1. Module Title</b>
Introduction to Programming

<b>2a. Module Leader</b>
Razvan-loan Dinita

<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>	None		

## LEARNING, TEACHING AND ASSESSMENT INFORMATION

### 6a. Module Description

Computers form a central part of everyday life and there is no indication that this is likely to change. Understanding how to program computers to perform specific tasks, such as navigation, inventory management, factory automation, financial transaction processing, and home automation, is a key skill in today's world. Assuming no prior programming experience, this module introduces you to the procedural programming paradigm.

You'll use industry-standard tools and techniques to design, implement, test and document simple procedural programs using a current programming language such as C, C++, or Python. The module focuses on helping you understand the key components of a computer program, providing a foundation for future modules that require structured programming ability.

The principles of good programming practice will be emphasised, and you'll be introduced to techniques required to develop software that is robust, efficient, meets user requirements, is written using of elegant, easy-to-read code, and is resilient from a cybersecurity perspective.

By the end of the module, you'll have sufficient mastery of a procedural programming language to allow you to design, implement and test simple programs. The skills taught in this module are intended to be directly transferable to the workplace, providing a solid foundation for a wide range of careers in computing.

### 6b. Outline Content

An introduction to:

- Control structures: sequence, selection, iteration, and Boolean algebra.
- Variables, constants, data types, and operators.
- Program design, program structure, testing/debugging methodologies, documentation.
- Functions/procedures, algorithms (such as searching and sorting).
- Simple data structures, such as arrays and records.
- File handling
- Principles of good programming practice such as reusability, maintainability, and intellectual property considerations.
- Secure coding through data validation, authentication and password management, access control, and exception handling.

### 6c. Key Texts/Literature

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

### 6d. Specialist Learning Resources

Students will need regular and frequent access to both hardware and software in order to develop their programming skills both on and off campus. A suitable programming environment and associated e-learning materials must be readily available. Open-access labs equipped with appropriate software and Internet access are required on the delivery campus. Students may download the programming environments and tutorial material used in this module for use at home.

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	Understand and describe the fundamental structures and syntax of programming in a popular procedural programming language.
2	Knowledge and Understanding	Understand and describe the fundamental concepts of Procedural Programming in a popular procedural programming language.
3	Knowledge and Understanding	Select and use appropriate techniques and tools to design solutions to a range of problems which can be coded in a procedural programming language.
4	Intellectual, practical, affective and transferrable skills	Create programs using appropriate syntax and structures in a procedural programming language for a range of problems and test them by selecting and using appropriate techniques.
5	Intellectual, practical, affective and transferrable skills	Create code which adheres to given guidelines of good programming practice suitable for the language used.
6	Intellectual, practical, affective and transferrable skills	Reflect on own programming skills, plan and record personal development as the foundation for lifelong learning.

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	24	1-6	2 x 1 hour lectures each week
Other teacher managed learning	24	1-6	2 hour lab each week
Student managed learning	252	1-6	On-line course materials supporting the lecture and tutorial series are to be provided via the Virtual Learning Environment
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	1-6	100 (%)	Fine Grade	30 (%)

Coursework assessment comprising one in-class test (equivalent to 4000 words) and one Logbook (equivalent to 1000 words).

**Assessment components for Element 010**

Component No.	Assessment Title	Submission Method	Weighting (%)	Components needed for Mark Calculation?
010/1	In-Class Test (90 minutes)		80 (%)	All
010/2	Logbook		20 (%)	

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]