Module Definition Form (MDF)

Module code: MOD002549	**Version:** 3 **Date Amended:** 02/Jul/2015

1. Module Title
Fundamentals of Computing

2a. Module Leader
Cristina Luca

2b. School
Cambridge School of the Creative Industries

2c. Faculty
Faculty of Arts, Humanities, Education and Social Sciences

3a. Level
3

3b. Module Type
Standard (fine graded)

4a. Credits
15

4b. Study Hours
150

5. Restrictions

Type	Module Code	Module Name	Condition
Pre-requisites:	None		
Co-requisites:	None		
Exclusions:	None		
Courses to which this module is restricted:	None		

LEARNING, TEACHING AND ASSESSMENT INFORMATION

6a. Module Description

This module provides an introduction to basic computer programming using a low level programming language (C), requiring no prior programming experience. Fundamental issues such as the structure of a program, syntax of simple statements, data types, functions, files, design and testing, and problem solving will be discussed. The students will use industrystandard tools and techniques to implement, test and document simple programs. The module will enable students to understand the main components of a high-level program, laying the foundation for subsequent modules requiring structured programming ability. It will emphasise the principles of good programming practice and introduce the techniques required to develop software that is robust, usable and efficient. By the end of the module, students should have sufficient mastery of the C programming language to allow them to design, implement and test simple programs. The material taught in the module is intended to form skills directly transferable to the workplace, giving a basic foundation to students who will be expected to apply programming skills in their later studies. Formative exercises will be carried out throughout the module, for early feedback.

6b. Outline Content

- Introduction to truth tables and Boolean algebra
- Introduction to programming in C: constants, variables, data types
- Structure and syntax of a C programme 39
- Simple data structures (arrays and files)
- Functions/procedures and algorithms (searching and sorting)
- Basic testing/debugging principles
- Software design notions (Pseudocode)
- Basic problem solving patterns (Divide and Conquer)
- Gradual development of a real world application

6c. Key Texts/Literature

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

6d. Specialist Learning Resources

PC laboratory.

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	Write, compile and debug programs in C language
2	Knowledge and Understanding	Understand the fundamentals of C programming in terms of data types/declarations, structure and syntax.
3	Intellectual, practical, affective and transferrable skills	Translate a sequence of steps expressed initially in pseudo-code into simple programmes.
4	Intellectual, practical, affective and transferrable skills	Translate a sequence of steps expressed initially in pseudo-code into simple programmes.

8a. Module Occurrence to which this MDF Refers				
Year	**Occurrence**	**Period**	**Location**	**Mode of Delivery**
2026/7	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	12	1-3	Lectures and Labs
Other teacher managed learning	12	1-3	Lectures and Labs
Student managed learning	126	1-4	Coursework and background reading, total 126 hours
TOTAL:	150		

9. Assessment for the above Module Occurrence

The assessment for some modules includes a presentation (see below for details on how this particular module is assessed). We recognise that presenting, especially to a large group, can cause significant anxiety or distress for some students. If this may affect you, please contact your Module Leader or the Disability and Neurodiversity Service as early as possible. We will work with you to consider appropriate reasonable adjustments. Depending on your individual needs, these might include, for example, presenting to a smaller group or presenting only to a member of academic staff.

Assessment No.	Assessment Method	Learning Outcomes	Weighting (%)	Fine Grade or Pass/Fail	Qualifying Mark (%)
010	Coursework	1-4	100 (%)	Fine Grade	30 (%)

Coursework assignment, which will test student's application of programming knowledge and skills; it will consist of a set of tasks to be solved through the design, implementation and testing of simple computer programs and the associated description of each solution (equivalent to 3000 words).

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]