Module Definition Form (MDF)

Module code: MOD008115	**Version:** 4 **Date Amended:** 23/Oct/2023

1. Module Title
Fundamentals of Computing

2a. Module Leader
Chinchu Babu

2b. School
SE: ARU College

2c. Faculty
Faculty of Science and Engineering

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:	N/A		

LEARNING, TEACHING AND ASSESSMENT INFORMATION

6a. Module Description

This module will provide you with an introduction to basic computer programming using a low level programming language (C), to which you require no prior programming experience.

You will learn about fundamental issues such as the structure of a program, syntax of simple statements, data types, functions, files, design and testing, and problem solving. You will use industry standard tools and techniques to implement, test and document simple programs.

This module will allow you to understand the main elements 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 this module, you should have sufficient mastery of the C programming language to allow you to design, implement and test simple programs. The material taught to you in this module is intended to form skills directly transferable to the workplace, giving you a basic foundation which will allow you to apply programming skills in your subsequent studies.

6b. Outline Content

• Learn to read and comment programs in C

• Structure and Syntax of a C program

• Introduction to programming in C: constants, variables, data types

• Sequence, selection and Iteration

• Simple data structures-Arrays

• Functions/procedures and algorithms (searching and sorting)

• Basic testing/debugging principles

• Software design notions (Pseudocode)

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/ Computers/ Laptops

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	Read, write, comment, 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

8a. Module Occurrence to which this MDF Refers				
Year	**Occurrence**	**Period**	**Location**	**Mode of Delivery**
2023/4	F01CAM	Trimester 1	ARU Cambridge Campus	Face to Face

8b. Learning Activities for the above Module Occurrence			
Learning Activities	**Hours**	**Learning Outcomes**	**Details of Duration, frequency and other comments**
Lectures	0	None	None
Other teacher managed learning	48	1-3	4 hours per week x 12 teaching weeks
Student managed learning	102	1-3	Pre and post session preparation, reading and research. Other tasks as detailed in module guide.
TOTAL:	150		

9. Assessment for the above Module Occurrence					
Assessment No.	**Assessment Method**	**Learning Outcomes**	**Weighting (%)**	**Fine Grade or Pass/Fail**	**Qualifying Mark (%)**
010	Coursework	1 2 3 4	50 (%)	Fine Grade	30 (%)
In-class Test up to (1.5 hours)					
Assessment No.	**Assessment Method**	**Learning Outcomes**	**Weighting (%)**	**Fine Grade or Pass/Fail**	**Qualifying Mark (%)**
011	Coursework	1 2 3 4	50 (%)	Fine Grade	30 (%)
In-class Test up to (1.5 hours)					

In order to pass this module, students are required to achieve an overall mark of 40%.
In addition, students are required to:
(a) achieve the qualifying mark for each element of fine graded assessment of as specified above
(b) pass any pass/fail elements