

Module Code	CSU11E03
Module Name	Computer Engineering I
ECTS Weighting¹	5 ECTS
Semester taught	Semester 2
Module Coordinator/s	Assistant Professor Lucy Hederman
Module Learning Outcomes with reference to the Graduate Attributes and how they are developed in discipline	On successful completion of this module, students should be able to: LO1. Analyse simple programming problems; LO2. Specify and design an algorithm to solve simple programming problems; LO3. Write C++ programmes to solve simple programming problems; LO4. Compile, run, test and debug C++ programmes; LO5. Select and use correctly appropriate control structures for specific programming sub-problems; LO6. Recognise the value of procedural abstraction and be able to use procedures to simplify programme design, hide detail and allow reuse of code; LO7. Use arrays where appropriate in the design and implementation of a programme; LO8. Predict the behaviour of a given C++ program that uses the concepts and constructs covered by the course. Graduate Attributes: levels of attainment To act responsibly - Introduced To think independently - Enhanced To develop continuously - Enhanced To communicate effectively - Not embedded

¹ [TEP Glossary](#)

Module Content

This module aims to equip students with the skills to design and develop simple imperative programs. It provides a solid grounding in algorithm design and programming techniques, in preparation for later courses that require programming. Topics include

- Introduction to computers and computing;
- Programming, compiling and running programmes;
- Basic C++ programmes; expressions, variables and data types, assignment;
- Selection and the IF-ELSE statement;
- Iteration, WHILE loops and FOR loops;
- Programme design process, algorithms and pseudocode;
- Advanced control flow: nested loops, nested IFs, the switch statement
- Procedural abstraction, functions in C+
- Arrays and array algorithms.

Teaching and Learning Methods

Recorded content, interactive lectures & programming laboratories.

Assessment Details² Please include the following: • Assessment Component • Assessment description • Learning Outcome(s) addressed • % of total • Assessment due date	Assessment Component	Assessment Description	LO Addressed	% of total	Week due
	Weekly programming	11 weekly programming exercises carried out during laboratory sessions.	All	20%	Each teaching week.
	Weekly mini-quizzes	Weekly online quizzes to be completed after review of each week's material.	LO1, LO2, LO5, LO6, LO7, LO8	5%	Each teaching week.
	Mid-semester test	Online real-time test.	LO1, LO2, LO5	10%	Week 6
	End-of-semester exam	2 Hour In-lab real-time exam.	LO1, LO2, LO3, LO5, LO6, LO7, LO8	65%	In exam period
Reassessment Requirements	3 hour In-lab real-time exam 100%				
Contact Hours and Indicative Student Workload²	Contact hours: 55 33 hours lectures; 22 hours laboratories.				
	Independent Study (preparation for course and review of materials): 20				
	Independent Study (preparation for assessment, incl. completion of assessment): 35				
Recommended Reading List	Main text for the course:				
	Required Texts C++ Programming: Program Design Including Data Structures, DS Malik, 6 th edition Cengage Learning, ISBN 978-1-133-52635-3				
	Or eBook version, http://www.cengagebrain.co.uk/shop/isbn/9781133526353				
Module Pre-requisite					
Module Co-requisite					

² [TEP Guidelines on Workload and Assessment](#)

Module Website	Blackboard
Are other Schools/Departments involved in the delivery of this module? If yes, please provide details.	School of Computer Science and Statistics
Module Approval Date	
Approved by	
Academic Year of Date	2024/2025