

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	<p>On successful completion of this module, students should be able to:</p> <p>LO1. Analyse simple programming problems;</p> <p>LO2. Specify and design an algorithm to solve simple programming problems;</p> <p>LO3. Write C++ programmes to solve simple programming problems;</p> <p>LO4. Compile, run, test and debug C++ programmes;</p> <p>LO5. Select and use correctly appropriate control structures for specific programming sub-problems;</p> <p>LO6. Recognise the value of procedural abstraction and be able to use procedures to simplify programme design, hide detail and allow reuse of code;</p> <p>LO7. Use arrays where appropriate in the design and implementation of a programme;</p> <p>LO8. Predict the behaviour of a given C++ program that uses the concepts and constructs covered by the course.</p> <p>Graduate Attributes: levels of attainment</p> <p>To act responsibly - Introduced</p> <p>To think independently - Enhanced</p> <p>To develop continuously - Enhanced</p> <p>To communicate effectively - Not embedded</p>

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: <ul style="list-style-type: none"> • Assessment Component • Assessment description • Learning Outcome(s) addressed • % of total • Assessment due date 	Assessment Component	Assessment Description	LO Addressed	% of total	Week due
	Mid-semester test	Online real-time test.	LO1, LO2, LO3, LO5	30%	Week 6
	End-of-semester test	Online real-time test.	LO1, LO2, LO3, LO5, LO6, LO7, LO8	40%	Week 12 or in exam period
	Coursework	10 weekly programming exercises carried out during laboratory sessions.	All	30%	Each teaching week.

Reassessment Requirements
Online real-time test 100% (TBC)

Contact Hours and Indicative Student Workload²	Contact hours: 55 33 hours pre-recorded material, lectures and quizzes; 22 hours laboratories.
	Independent Study (preparation for course and review of materials): 20
	Independent Study (preparation for assessment, incl. completion of assessment): 25

Recommended Reading List
Main text for the course:

Required Texts
C++ Programming: Program Design Including Data Structures, DS Malik, 6th 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

Module Website Blackboard

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

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 Start Year

September 28th 2021

Academic Year of Date

2021/2022