Module Code	CSU11022				
Module Name	Introduction to Computing II				
ECTS Weighting ¹	5 ECTS				
Semester taught	Semester 2				
Module Coordinator/s	Jonathan Dukes				
Module Learning Outcomes	 On successful completion of this module, students will be able to: LO1. Describe the characteristics, structure and operation of a basic computer system, including the execution of subroutines and the interface between the processor and external devices; LO2. Translate between high-level programming language constructs, including fundamental data structures and subroutines, and their assembly language equivalents; LO3. Design, construct, document and test assembly language programs to solve small-scale problems of moderate complexity by decomposing the problems into smaller parts and implementing solutions consisting of one or more assembly language programs that can interact with simple external devices. 				
Module Content	 This module continues directly from CSU11021 and examines the structure and behaviour of computer systems in greater depth. In particular, this module introduces students to the implementation of simple data structures (stacks, multi-dimensional arrays, composite data types), subroutines (including parameter passing conventions), exceptions, interrupts and basic I/O at the machine level. Students are given opportunities throughout the module to reinforce their problem solving, programming and written communication skills by designing, implementing, documenting and testing solutions to programming problems of increasing complexity. Problem decomposition is strongly encouraged. 				
Teaching and Learning Methods	Lectures are used to introduce key concepts and provide worked examples. Every fortnight, each student participates in a tutorial to further explore each topic. In the tutorials, students work in groups of up to six on a set of exercises, using whiteboards to explore solutions, with guidance and feedback from teaching staff. Students work in pairs on four sets of lab exercises and are given two weeks to work on each set, beginning the exercise in the scheduled labs and completing them outside scheduled hours. At the end of each two-week cycle, each pair of students demonstrates their work to teaching staff and receives feedback during the lab. Finally, a substantial mid-term assignment provides students with an opportunity to work individually on a larger-scale problem.				

¹ TEP Glossary

Assessment Details ²	Assessment Component	Brief Description	Learning Outcomes Addressed	% of total	Week set	Week due		
	Examination	2 hour examination	LO1, LO2, LO3,	70%	n/a	n/a		
	Lab 1	Stacks and subroutines	LO1. LO2	2.5%	3	4		
	Lab 2	Multi-dimensional arrays	L01, L02	2.5%	5	6		
	Assignment	Design, implement, test and document a program of moderate complexity	LO2, LO3	20%	5	8		
	Lab 3	Floating point representation	LO1, LO2	2.5%	9	10		
	Lab 4	I/O, exceptions and interrupts	LO4	2.5%	11	12		
Reassessment Details	Examination	(2 hours, 100%)						
Contact Hours and Indicative Student Workload	Contact Hours (scheduled hours per student over full module), broken down by:					urs		
	lecture					22 hours		
	tutorial or seminar					rs		
	other					rs		
	Independent study (outside scheduled contact hours), broken down by:					78 hours		
	preparation for classes and review of material (including preparation for examination, if applicable)					urs		
	completion of assessments (including examination, if applicable)					urs		
	Total Hours					ours		
Recommended Reading List	William Hohl, "ARM Assembly Language: Fundamentals and Techniques", CRC Press, 2009.							
	Steve Furber, "ARM System-on-Chip Architecture", 2nd edition, Addison-Wesley Professional, 2000. [suggested further reading]							
	Andrew Sloss, Dominic Symes and Chris Wright, "ARM System Developer's Guide: Designing and Optimizing System Software", Morgan Kaufmann, 2004. [suggested further reading]							
Module Pre-requisites	Prerequisite modules: CSU11021							
	Other/alternative non-module prerequisites: A basic working knowledge of the ARM instruction set and some familiarity with at least one high level programming language.							
Module Co-requisites								
Module Website	Blackboard / mymodule.tcd.ie							
Last Update	28/06/2019 by Jonathan Dukes							

² TEP Guidelines on Workload and Assessment