MODULE DESCRIPTOR

Module Title

Computer Engineering

Reference	EN2541	Version	2
Created	March 2018	SCQF Level	SCQF 8
Approved	September 2017	SCQF Points	15
Amended	July 2018	ECTS Points	7.5

Aims of Module

To provide the student with the ability to describe the operation of microcomputer systems and develop, test and document structured software in a high-level language.

Learning Outcomes for Module

On completion of this module, students are expected to be able to:

1	Describe the structure of a microcomputer system and explain its principles of operation.
2	Apply knowledge of hardware software in the use of a microprocessor development system.
3	Demonstrate the ability to use a development system for a high level programming language and create programs using it.
4	Design, code, test and document modular structured programs in a high-level programming language to prescribed standards and specifications.
5	Explain the characteristics of a typical programming language, algorithms and data structures and the process of software development.

Indicative Module Content

Microcomputer systems: operation of the CPU, registers, ALU, control unit, address, data and control buses, memory, input/output ports, system clock and timing, the fetch-execute cycle and memory maps. Machine instructions: opcodes, operands and addressing modes, data transfer, arithmetic and logical operations, control structures, flags, subroutines. Software development: algorithms, source and object code, compilers, the edit-compile-execute cycle, software design, testing, standards and documentation. Syntax of a high-level language: constants and variables, data types, pointers, arrays and data structures; program expressions and statements, input and output, selection and repetition control structures; modular programming, library and user functions, parameter passing, macros.

Module Ref:	EN2541 v2

Module Delivery

This module is delivered using a structured programme of lectures, tutorials and laboratory exercises supplemented by directed reading and student-centred learning.

Indicative Student Workload	Full Time	Part Time
Contact Hours	60	36
Non-Contact Hours	90	114
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	150	150
Actual Placement hours for professional, statutory or regulatory body		

ASSESSMENT PLAN

If a major/minor model is used and box is ticked, % weightings below are indicative only.

Component 1

Type:	Coursework	Weighting:	25%	Outcomes Assessed:	2, 3
Description:	Logbook of practical activities.				

Component 2

Type:	Practical Exam	Weighting:	25%	Outcomes Assessed:	4
Description:	Open book programming examination.				

Component 3

Type:	Examination	Weighting:	50%	Outcomes Assessed:	1, 5
Description:	Closed book examination.				

MODULE PERFORMANCE DESCRIPTOR

Explanatory Text

The module grade is calculated as the weighted average of the component marks. To pass the module students must get at least a grade D overall AND 35% or more in Component 3.

Module Grade	Minimum Requirements to achieve Module Grade:
A	70% - 100%
B	60% - 69%
C	50% - 59%
D	40% - 49%
E	35% - 39%
F	0% - 34%
NS	Non-submission of work by published deadline or non-attendance for examination

Module Requirements

Prerequisites for Module	None.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

1. WEERT, P.V. and GREGOIRE, M., 2016. C++ standard library quick reference. Berkeley, CA: Apress.
2. HORTON, I., 2014. Beginning C++. Berkeley, CA: Apress.
3. SUTHERLAND, B., 2015. C++ recipes: a problem-solution approach. Berkeley, CA: Apress.