This Version is No Longer Current
The latest version of this module is available here

MODULE DESCRIPTOR

Module Title

Languages and Compilers			
Reference	CMM306	Version	2
Created	December 2020	SCQF Level	SCQF 11
Approved	April 2017	SCQF Points	15
Amended	March 2021	ECTS Points	7.5

Aims of Module

This module will provide students with the theory, knowledge and skills that will provide them with a critical understanding of programming language design and facilitate in the implementation of compilers.

Learning Outcomes for Module

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

1 Analyse and apply language syntax specifications and semantics.

2 Evaluate the structure and phases of a range of compilation processes and techniques.

3 Critically evaluate the relationships between target platform architecture, compilers and programming language design.

4 Create and apply lexical and syntax analysis of a given language using appropriate techniques.

5 Construct a code generating compiler for a given source language and target architecture.

Indicative Module Content

This module will cover the theoretical knowledge and practical skills required to understand and construct conventional compilers for bespoke programming languages. The module will cover 4 main topics in language compiler and development. 1) Language Specification - Syntax and semantics, language grammar, derivation sequences. 2) Compiler Architecture - components, phases and general structures of compilers, including FSM approach and phase interactions 3) Analysis - Lexical, Syntax and semantic analysis as a compiler phase. Recursive descent compilation. Error detection 4) Code / Artefact Generation - generation of target platform code or artefact.

Module Delivery

Key concepts are introduced and illustrated through lectures and directed reading. The understanding of students is tested and further enhanced through interactive tutorials. In the laboratories, the student will progress through a sequence of exercises to develop sufficient knowledge and skills in language design and compiler development.

Indicative Student Workload	Full Time	Part Time
Contact Hours	44	N/A
Non-Contact Hours	106	N/A
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	150	N/A
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:	Examination	Weighting:	50%	Outcomes Assessed:	1, 2, 3
Description:	A written exam.				

Component 2

Type:	Coursework	Weighting:	50%	Outcomes Assessed:	4, 5
Description:	This coursework comprises the implementation of a complete, working compiler.				

MODULE PERFORMANCE DESCRIPTOR

Explanatory Text

There are two assessment componets: a written examination worth 50% and a coursework worth 50%. The minimum grade to pass the module is D.

		Examination:						
		A	**B**	**C**	**D**	**E**	**F**	**NS**
Coursework:	**A**	A	A	B	B	C	E	
	B	A	B	B	C	C	E	
	C	B	B	C	C	D	E	
	D	B	C	C	D	D	E	
	E	C	C	D	D	E	E	
	F	E	E	E	E	E	F	
	NS	Non-submission of work by published deadline or non-attendance for examination						

Module Requirements

Prerequisites for Module	CM3113 Concurrent Programming or equivalent.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

1 Watt, D.A., Brown, D.F. and Brown, D., 2000. Programming language processors in Java: compilers and interpreters. Pearson Education.

2 Lam, M., Aho, A.V., Sethi, R. and Ullman, J.D., 2013, Compilers: principles, techniques and tools. 2nd edition, Addison Wesley

3 Abelson, H. and Sussman, G.J., 1996. Structure and interpretation of computer programs. The MIT Press.