# ROBERT GORDON
## UNIVERSITY ABERDEEN

**This Version is No Longer Current**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

## MODULE DESCRIPTOR

### Module Title

Languages and Compilers

| Reference | CM4106 | Version | 3 |
|---|---|---|---|
| Created | June 2022 | SCQF Level | SCQF 10 |
| Approved | April 2017 | SCQF Points | 15 |
| Amended | July 2022 | 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 simple compilers.

### Learning Outcomes for Module

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

| 1 | Understand and interpret language syntax specifications and semantics. |
|---|---|
| 2 | Explain the structure and phases of a conventional compilation process. |
| 3 | Critically evaluate the relationships between target platform architecture, compilers and programming language design. |
| 4 | Design and implement lexical and syntax analysis of a given language using appropriate techniques. |
| 5 | Construct a code generating compiler for a simple source language and a simple 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 | 30 | N/A |
| Non-Contact Hours | 120 | 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: A coursework which involves the implementation of a complete, working compiler.

## MODULE PERFORMANCE DESCRIPTOR

### Explanatory Text

There are two assessment components: 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** |
| | **A** | A | A | B | B | C | E | |
| | **B** | A | B | B | C | C | E | |
| | **C** | B | B | C | C | D | E | |
| Coursework: | **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. |