

Liverpool John Moores University

Title: COMPILER DESIGN
Status: Definitive
Code: **5121COMP** (121248)
Version Start Date: 01-08-2021

Owning School/Faculty: Computer Science and Mathematics
Teaching School/Faculty: Computer Science and Mathematics

Team	Leader
Reino Niskanen	Y

Academic Level: FHEQ5 **Credit Value:** 20 **Total Delivered Hours:** 57
Total Learning Hours: 200 **Private Study:** 143

Delivery Options

Course typically offered: Semester 1

Component	Contact Hours
Lecture	33
Practical	11
Seminar	11

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Report	AS1	Report on practical low level code generation	40	
Exam	AS2	Examination	60	2

Aims

To introduce the concepts of languages and grammars leading to the principles of Compiler Design.

Learning Outcomes

After completing the module the student should be able to:

- 1 Understand the principles of recognisers for programming languages
- 2 Illustrate efficient techniques for semantic analysis
- 3 Describe common programming paradigms
- 4 Represent the major constructs of programming languages as low level code.

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Code generation	4			
Exam	1	2	3	

Outline Syllabus

Languages and grammars

Regular expressions, context-free grammars, BNF.

Parsing techniques.

Attribute grammars

Translation schemes

Type inference

Symbol tables.

Code generation

Translation to intermediate code

Register allocation

Compiler optimization.

Programming Paradigms

Object-oriented and functional programming languages.

Learning Activities

Lectures will describe the main elements of theory, whilst laboratories and seminars will be used to examine the practical aspects of the module and illustrate the main concepts.

Notes

This module will introduce the theory of programming languages and associated compiler design. It will cover the rationales for various languages and paradigms and provide in depth materials on the fundamentals of designing compilers for particular languages.