

# Programming Language Theory

## Module Information

2022.01, Approved

### Summary Information

Module Code	5129COMP
Formal Module Title	Programming Language Theory
Owning School	Computer Science and Mathematics
Career	Undergraduate
Credits	20
Academic level	FHEQ Level 5
Grading Schema	40

### Teaching Responsibility

LJMU Schools involved in Delivery
Computer Science and Mathematics

### Learning Methods

Learning Method Type	Hours
Lecture	33
Practical	22

### Module Offering(s)

Display Name	Location	Start Month	Duration Number Duration Unit
SEP-CTY	CTY	September	12 Weeks

### Aims and Outcomes

Aims	The module's aim is to provide an introduction to the concepts behind programming languages, along with an explanation of the underpinnings of programmable machines. It will also discuss and demonstrate a variety of programming languages across both Imperative and Declarative paradigms
------	--

**After completing the module the student should be able to:**

**Learning Outcomes**

Code	Number	Description
MLO1	1	Explain the key concepts in specifying and evaluating a programming language.
MLO2	2	Apply appropriate formal methods to specify a programming language
MLO3	3	Design and implement an interpreter/compiler for a simple imperative programming language
MLO4	4	Appraise Imperative and Declarative programming paradigms as an appropriate mechanism for a variety of problem domains

**Module Content**

Outline Syllabus	Abstract views of program execution:FSMs, the interlock machineTuring Machines; the computing machine; addition of configurable memoryDefining and processing a language:Fundamental Language Theory: Grammars and SyntaxFundamental Compiler Theory: Lexical, Syntactical and Semantic AnalysisParse/syntax trees and bindingLanguage Paradigms:Imperative vs Declarative LanguagesState transformations vs. Referential transparencyHow vs Why & why is this important?
Module Overview	
Additional Information	This module combines theory and practical work to familiarise a student with the fundamentals of programming languages and their compilation / interpretation for execution; culminating with the student specifying and designing their own basic imperative language. It will make use of programming-related skills from previous modules; particularly functional decomposition and basic data design. It will, in parts, reinforce and present alternative use cases for materials covered in Data Structures. The module will assume that the student is already with some fundamentals of imperative programming, namely:-variable declaration, state modification and scope. -how to design and write software that correctly uses sequential execution, branching, iteration and function-calling.

**Assessments**

Assignment Category	Assessment Name	Weight	Exam/Test Length (hours)	Module Learning Outcome Mapping
Technology	Basic language, interpreter	60	0	MLO1, MLO2, MLO3, MLO4
Centralised Exam	Examination	40	2	MLO1, MLO2

**Module Contacts**

**Module Leader**

Contact Name	Applies to all offerings	Offerings
Somasundaram Ravindran	Yes	N/A

**Partner Module Team**

Contact Name	Applies to all offerings	Offerings