6CCS3CFL Compilers and Formal Languages

[KEATS MODULE PAGE](https://keats.kcl.ac.uk/course/view.php?id=77779)

**Credit value:** 15

**Semester**: 1

**Aims and Learning Outcomes**

To explain the techniques behind compilers, lexers and parsers. To introduce the mathematical formalisms of regular expressions, context-free grammars, and to show their applications to computer languages.

To generate code for low level machine languages and to illustrate compiler techniques.

On successful completion of this module, students will:

Have learned:

* CFL1. How to use regular expressions for lexing programs and data.
* CFL2. How to design grammars for parsing programs.
* CFL3. How to implement a parser and an interpreter.
* CFL4. How to implement a compiler.

Be able to:

* CFL5. Implement the central components of a small compiler (including lexer, parser and code-generator).

**Syllabus**

1. Formal languages

 alphabets, strings, languages

 mathematical background, sets and relations, proofs

1. Regular expressions

 regular expression matching

 derivatives of regular expressions

 lexing and token generation

1. Finite Automata

 Deterministic Finite Automata (DFA)

 Non-Deterministic Finite Automata (NFA)

 equivalence of DFA, NFA and regular expressions

1. Parsing Techniques

 context-free grammars

 parser combinators

1. Interpreters and Compilers:

 code generation for the WHILE-language

 compilation to the Java Virtual Machine and the LLVM-IR

 compiling a functional language

 compiler optimisation techniques

**Assessment**

30% written examination (2 hours)

15% midterm exam

45% coursework (5 assignments)

10% weekly engagement