

CS3230 Programming Language Implementation

Level: 3 Credits: 10 Teaching Period: 2 Module Tutor: Dr EF Elsworth

Aims

To provide a general understanding of the challenges faced in programming language implementation, and how those challenges can be overcome, having regard to both scientific principle and practical considerations. To provide a software engineering illustration by studying the design of the substantial software systems needed for language implementation. To “bridge the gap” between study in software and computer systems areas.

Content

Structure of programming language systems: compilers, interpreters, phases of compilation, organisation into passes, interaction with users.

Lexical analysis: regular expressions, finite state automata, coding techniques.

Syntactic specification: BNF, EBNF, limitations of context-free syntax.

Parsing: recursive descent method. Representation of abstract syntax tree.

Static semantics/context-sensitive syntax. Symbol table organisation and use.

Representation of values: primitive and composite.

Run time organisation: static and dynamic store allocation, stacks, support for procedures and functions, non-local environment, OOPL requirements, heap allocation, garbage collection.

Code generation: organisation, design, optimisation, interface with operating environment.

A substantial case study (implementation of a simple language with a virtual machine target) will be explored during tutorials; associated practical work will provide an opportunity to acquire an enhanced understanding of language implementation techniques.

Teaching

Lectures: 20 hours (2 hours/week)

Tutorials: 10 hours (1 hour/week)

Assessment

Written exam: 100% (3 hours, January)

The exam will include questions related to the case study and associated practical techniques (*)

Module outcomes

<i>What the student should gain from successful completion of the module</i>	<i>Teaching/Learning Methods</i>	<i>Assessment Methods</i>
<i>Knowledge and Understanding</i>		
The general organisation of programming language systems The scientific and technical principles that support programming language implementation How typical computer hardware supports realisation of programming language constructs	Lectures supported by appropriate reading	Exam
<i>Intellectual Skills</i>		
Able to select and apply appropriate design and development techniques in the context of example problems and a more substantial case study	Lectures and tutorials	Exam – see (*) above
<i>Professional/Subject-Specific Skills</i>		
Able to apply relevant technical specification notations (eg EBNF, regular expressions)	Tutorials and practical work	Exam – see (*) above
<i>Transferable Skills</i>		
Enhanced skills in problem analysis and application of formalised methods	Practical work	Exam – see (*) above

Learning resources

D Watt & D Brown: Programming Language Processors in Java, Prentice Hall, 2000

M L Scott: Programming Language Pragmatics, Morgan-Kaufmann, 2000

Other study requirements to take this module

CS1410 Java Program Development

CS1030 Computer Systems Principles, or equivalent background