

<a href="#">Module Title</a>	Compiling Techniques	<a href="#">School</a>	Mathematical & Computer Sciences				<a href="#">On or Off-Campus</a>	On Campus	
<a href="#">Module Co-ordinator</a>	Andrew Ireland/Rob Pooley	<a href="#">SCQF Level</a>	9	<a href="#">Module Code</a>	F23PF	<a href="#">Term</a>	2	<a href="#">Credits</a>	10

1. <a href="#">Pre-requisites</a>	F22AO Data structures and Algorithms 1 F22AW Foundations of Computing F22PQ System Level Programming		
2. <a href="#">Linked Modules (specify if synoptic)</a>	None		
3. <a href="#">Excluded Modules</a>			
4. <a href="#">Replacement Module</a>	Code: Date Of Replacement:	5. <a href="#">Availability as an Elective</a>	Yes No <input checked="" type="checkbox"/>
6. <a href="#">Degrees for which this is a core module</a>	BSc Computer Science, MEng Software Engineering		

7. [Aims](#)

To give an introduction and an appreciation of the principles of compiler construction.

8. [Syllabus](#)

Compiler architectural issues. Formal grammars: production rules, grammar classification, derivations. Lexical analysis: regular expressions [Lex], symbol tables. Syntax analysis: writing grammars, top-down parsing [recursive descent], bottom-up [YACC]. Code generation. Syntax directed translation schemes. Internal forms. Control statements.

9. [Learning Outcomes](#)[Subject Mastery](#)*Understanding, Knowledge and Subject-Specific Skills*

- ◆ Broad and integrated knowledge and understanding of the principal components and functionality of a compiler
- ◆ Critical understanding of the role of formal grammars in the construction of compilers
- ◆ Detailed knowledge of some of the problems and solutions that can arise when implementing a language parser
- ◆ Critical understanding of the role and practice in the use of tools within the compiler construction process
- ◆ Critical understanding of the use of formal specification of semantics in relation to code generation
- ◆ Practice in semantic checking and code generation
- ◆ Broad Knowledge of code optimisation at high and low levels within a compiler
- ◆ Broad understanding of the issues relating to portability of compilers

<u>Module Title</u>	Compiling Techniques	<u>School</u>	Mathematical & Computer Sciences				<u>On or Off-Campus</u>	On Campus	
<u>Module Co-ordinator</u>	Andrew Ireland/Rob Pooley	<u>SCQF Level</u>	9	<u>Module Code</u>	F23PF	<u>Term</u>	2	<u>Credits</u>	10

<u>Personal Abilities</u>	<b><i>Cognitive skills, Core skills and Professional Awareness</i></b>				
	<ul style="list-style-type: none"> <li>◆ Practice in making reasoned judgements about engineering decisions when presented with a range of technical solutions</li> <li>◆ Critical analysis of the trade-offs between complementary approaches to computational problems</li> <li>◆ Practice in the use of a range of previously acquired skills</li> </ul>				
<b>10. <u>Assessment Methods</u></b>				<b>11. <u>Re-assessment Methods</u></b>	
<b>Method (e.g. exam)</b>	<b>Length of Examination</b>	<b>Weighting (%)</b>	<b>Synoptic modules?</b>	<b>Method (e.g. exam)</b>	<b>Length of Examination</b>
Exam	2 hours	80%		Exam	2 hours
Coursework		20%			
<b>12. <u>Date and Version</u></b>					
<b>Date of Proposal</b>	7/12/04	<b>Date of Implementation</b>	October 2004	<b>Version Number</b>	2.0