

Module Title	Formal Specification	School	Mathematical and Computer Sciences				On or Off-Campus	On	
Module Co-ordinator	Murdoch Gabbay, Greg Michaelson	SCQF Level	9	Module Code	F29FS	Semester	2	Credits	15

1. Pre-requisites	Discrete Maths							
2. Linked Modules (specify if synoptic)								
3. Excluded Modules	None							
4. Replacement Module	Code:	5. Availability as an Elective		Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>	
	Date Of Replacement:							
6. Degrees for which this is a core module	Mandatory module for BSc Computer Science & MEng Software Engineering							
7. Aims								
To introduce students to specification of programs in a formal logical language (Z), and demonstrate the path from this to programs in a programming language (ML).								
8. Syllabus								
<ul style="list-style-type: none"> ◆ Reprise of logic and proof. ◆ Basic notions of set theory. ◆ Introduction to Z notation ◆ Relations and functions ◆ Schemas and specification structure ◆ Introduction to SML: atomic types, structured types, pattern matching, recursion, higher order constructs, exceptions ◆ Representing Z constructs in SML: predicates, sets, relations, functions ◆ Converting Z schema to SML: invariants as assertions, preconditions as guards, postconditions as actions ◆ Schema-based prototype testing 								

Module Title	Formal Specification	School	Mathematical and Computer Sciences				On or Off-Campus	On	
Module Co-ordinator	Murdoch Gabbay, Greg Michaelson	SCQF Level	9	Module Code	F29FS	Semester	2	Credits	15

9. Learning Outcomes (HWU Core Skills: Employability and Professional Career Readiness)

Subject Mastery	Understanding, Knowledge and Cognitive Skills Scholarship, Enquiry and Research (Research-Informed Learning)		
	<ul style="list-style-type: none"> ◆ Accurate understanding of the syntax and informal semantics of Z. ◆ Given a real world example, create a formal specification to model it. ◆ Reason with the specification, once produced. ◆ Given a specification, discuss what it models. ◆ Understand ML: polymorphic types, recursive functions, higher order functions, pattern matching, structured types. ◆ Given a specification, interpret it as operational requirements of a program. ◆ Given a specification, refine it to a program satisfying that specification. 		
Personal Abilities	Industrial, Commercial & Professional Practice Autonomy, Accountability & Working with Others Communication, Numeracy & ICT		
	<ul style="list-style-type: none"> ◆ Understand the importance of specification and its distinction from implementation ◆ Develop mathematical specification skills ◆ Appreciation of the role of formal techniques in software development. ◆ Ability to deploy declarative abstraction in software development. ◆ Awareness of limitations of formal techniques. 		

10. Assessment Methods				11. Re-assessment Methods	
Method	Duration of Exam (if applicable)	Weighting (%)	Synoptic modules?	Method	Duration of Exam (if applicable)
Exam	2 hours	75%		Exam	2 hours
Coursework/Small application		25%			

12. Date and Version							
Date of Proposal	7 November 2007	Date of Approval by School Committee	December 2007	Date of Implementation	September 2008	Version Number	1