

Module Title	Software Design	School	MACS				On or Off-Campus	On	
Module Co-ordinator	Andrew Ireland and Monica Farrow	SCQF Level	8	Module Code	F28SD	Semester	2	Credits	15

1. Pre-requisites	F27SA – Software Development 1 OR <i>equivalent</i>		
2. Linked Modules (specify if synoptic)			
3. Excluded Modules			
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; BSc Information Systems & MEng Software Engineering		
7. Aims	<ul style="list-style-type: none"> ◆ To provide an understanding of the challenges associated with the software life-cycle, focusing on the processes and techniques which promote the design of high quality software. ◆ To instil an understanding of the concepts and benefits of rigorous design. ◆ To equip participants with a broad understanding of software design methods and processes. ◆ To equip participants with an in-depth knowledge of UML notation ◆ To instil an understanding of design patterns and their value ◆ To provide a broad understanding of verification and validation issues associated with software engineering. ◆ To equip CS participants with an in-depth understanding of analysis techniques, i.e. static analysis and dynamic analysis. ◆ To instil IS participants with a broad understanding of the information systems lifecycle, and an in-depth understanding of feasibility studies. ◆ To provide participants with an understanding of software design challenges in the real-world via guest speakers from industry. 		

Module Title	Software Design	School	MACS				On or Off-Campus	On	
Module Co-ordinator	Andrew Ireland and Monica Farrow	SCQF Level	8	Module Code	F28SD	Semester	2	Credits	15

8. Syllabus

- ◆ Motivations for design and the need for design processes and methodologies.
- ◆ Software process models, e.g. waterfall model, Boehm's spiral model, evolutionary development, agile development.
- ◆ Design methods and processes, i.e. architectural design, function-oriented design, object-oriented design, component-based design.
- ◆ Object-oriented design using UML notation & CRC cards: use case diagrams, noun verb analysis, class diagrams, sequence diagrams, communication diagrams, state machine diagrams, activity diagrams.
- ◆ An introduction to patterns and anti-patterns
- ◆ Software testing and analysis: a strategy for dynamic testing; test case design; assertion based testing; functional testing; structural testing; hybrid testing; auto unit testing; reviews & inspections; metrics; flow analysis; formal methods (CS stream)
- ◆ Open source vs closed source and means of assessing which is appropriate to a project – Boehm's ADA and other methods
- ◆ Documentation practice and standards in software engineering and information systems
- ◆ Information systems lifecycle focussing on feasibility studies. Evaluating potential projects by considering operational, technical, schedule and economic feasibility. Cost/benefit analysis. (IS stream).

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

Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills</i> <i>Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To demonstrate a critical understanding of modern software engineering practice and be able to evaluate the strengths and weaknesses of current software engineering methods and technique ◆ To be able to choose appropriate metrics to measure software quality and quantity in a modern software engineering environment ◆ To understand and demonstrate object oriented design techniques, including the use of UML notation ◆ To understand the role of a feasibility study, and to appreciate social and technological aspects of information systems (IS stream)
Personal Abilities	<p><i>Industrial, Commercial & Professional Practice</i> <i>Autonomy, Accountability & Working with Others</i> <i>Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Take responsibility for own work and exhibit critical reflection on development process. (PDP) ◆ To be able to use appropriate methods and standards for practice and documentation in software engineering and information systems. ◆ Demonstrate evidence based approaches to problem solving. ◆ Use a range of numerical and graphical skills in evaluating and communicating ideas, as well as measuring progress toward achieving goals.

Module Title	Software Design	School	MACS				On or Off-Campus	On	
Module Co-ordinator	Andrew Ireland and Monica Farrow	SCQF Level	8	Module Code	F28SD	Semester	2	Credits	15

10. Assessment Methods				11. Re-assessment Methods			
Method	Duration of Exam (if applicable)	Weighting (%)	Synoptic modules?	Method	Duration of Exam (if applicable)		
Exam	2hr	80%		Exam	2hr		
Coursework		20%					
12. Date and Version							
Date of Proposal	29/06/09	Date of Approval by School Committee		Date of Implementation	Jan 2010	Version Number	2