MODULE SPECIFICATION

Part 1: Information			
Module Title	Analysis and Verification of Concurrent Systems		
Module Code	UFCFYN-15-M	Level	Level 7
For implementation from	2018-19		
UWE Credit Rating	15	ECTS Credit Rating	7.5
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Contributes towards	Cyber Security [Sep][FT][Frenchay][1yr] MSc 2018-19		
Module type:	Standard		

Pre-requisites	None
Excluded Combinations	None
Co- requisites	None
Module Entry requirements	None

Part 2: Description

Overview: This module will introduce you to the basic theory and principles of concurrent programming, their use in designing computer programs, with a special focus on formal description, analysis, and correctness of concurrent systems.

Educational Aims: In particular, you will study the safety and liveness properties of concurrent algorithms and protocols or policies, proving properties using assertional reasoning and model checking techniques. This module will focus on, and allow you to examine conventional testing methods as well as formal approaches, exploring the emerging challenges and limitations. This module will highlight how formal approaches enable reasoning from logical or mathematical specifications of the behaviours of concurrent processes, and how they offer rigorous proofs that all system behaviours meet some desirable property.

Outline Syllabus: Theory and concepts of concurrent systems: concurrency primitives for shared memory and distributed implementations of concurrency, and their use in solving some common problems in concurrent programming. (A)

Testing and formal verification techniques: concepts of static vs. dynamic testing, traditional testing vs. formal verification, test case design, path testing, statement and branch coverage, formal specification and verification using the model checking technique. (A, B)

Formal specification: concepts of formal specifications expressed in languages with formally defined syntax and semantics. Model-based specifications, specifying system behaviour by constructing a model in terms of welldefined mathematical constructs. Property-based specifications, specifying system behaviour in terms of properties that must be satisfied considering Linear Temporal Logic (LTL) and Computation Tree Logic (CTL). (A, B)

Tool support and case study: throughout the module, an emphasis will be given to formal approaches, tool support and the application of techniques in domains such as computer/mobile security. The module will provide practical experience in using modern verification tools such as NuSMV and/or Java Pathfinder. (B)

Teaching and Learning Methods: The module is delivered through a combination of formally scheduled sessions and independent learning. The scheduled learning includes lectures, tutorials, demonstrations and practical sessions. The independent learning will constitute the remaining study time and will be spent on self-directed study, support of the coursework, and preparation for the summative coursework demonstration.

Practical exercises will allow the students to gain familiarisation with the tools and techniques required for the specification and verification of safety and liveness properties of concurrent systems.

Part 3: Assessment

The summative assessment strategy for this module is a combination of written examination and coursework assignment:

Component A: the written examination (2 hours) will assess the students' knowledge, understanding and application of the concepts, problems and techniques associated with concurrent programming. It will also assess the students' application of formal techniques to the specification of system properties.

Component B:
The coursework assignment assesses, via a case study, the students' application of practical skills in modelling and reasoning about the critical properties of concurrent systems. Students will be required to submit a portfolio containing documents for the modelling and verification of the system's properties, including the system encoding, a log of the analysis, and the results of verification. The case study will be related to automated analysis and verification of security protocols.

Students will have the opportunity for formative feedback during practical lab sessions.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component B		40 %	Case study portfolio
Examination - Component A	✓	60 %	Written examination (2 hours)
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component B		40 %	Case study portfolio

Examination - Component A	✓	60 %	Written examination (2 hours)

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will be able to:

	Module Learning Outcomes
MO1	Demonstrate in-depth understanding of the concepts, problems, and techniques of concurrent programming
MO2	Evaluate approaches and techniques to develop safe and secure concurrent systems
MO3	Distinguish, contrast, and apply the concept of traditional testing and formal verification
MO4	Demonstrate the application of formalisms to specify system properties using temporal logics like Linear-time Temporal Logic (LTL) and Computation Tree Logic (CTL)
MO5	Use tools and analysis techniques to study and reason about critical properties of the concurrent systems, including security protocols

Contact Hours

Contact Hours	
Independent Study Hours:	
Independent study/self-guided study	114
Total Independent Study Hours:	114
Scheduled Learning and Teaching Hours:	
Face-to-face learning	36
Total Scheduled Learning and Teaching Hours:	36
Hours to be allocated	150
Allocated Hours	150

Reading List

The reading list for this module can be accessed via the following link:

https://uwe.rl.talis.com/modules/ufcfyn-15-m.html