

CS347-15 Fault Tolerant Systems

20/21

Department

Computer Science

Level

Undergraduate Level 3

Module leader

Matthew Leeke

Credit value

15

Module duration

10 weeks

Assessment

Multiple

Study location

University of Warwick main campus, Coventry

Description

Introductory description

The module concentrates on the principles and technologies that can be applied in the design, development and measurement of fault tolerance under varied assumptions. You will have the opportunity to analyse, design and write software based on state-of-the-art approaches in dependable systems.

Module aims

The aim of the module is to provide you with a knowledge of advanced issues and concepts in the design, implementation and evaluation of fault-tolerant systems.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

General: Fault, error, failure, fault transformation process. Implications of coverage on dependability, specifications, methods to achieve dependability.

Middleware: Protocols for synchronous distributed systems (leader election, consensus, clock synchronisation, Byzantine agreement and FDIR).

Protocols and abstractions for asynchronous distributed systems, including logical and vector

clocks, broadcast (best-effort, unordered reliable, ordered reliable), failure detectors, global predicate detection in fault-free and faulty systems.

Learning outcomes

By the end of the module, students should be able to:

- General: Understand dependability attributes, threats and means. Understand the differences between fault, error and failure. Discuss the process by which a fault eventually causes a system failure. Understand the link between fault model and the corresponding dependability mechanisms. Introduction of terms such as fail-safe, fail-operational, fail-stop, etc. Concepts such as fault tree, FMECA, FMEA, etc.
- HW/System: Calculate reliability of a system. Use of tools for reliability modelling. Design of dependable HW.
- Middleware: Understand critical functions such as clock synchronisation, consensus, FDIR protocols, etc. Understand Byzantine failures and its impact on system complexity. Introduction to asynchronous message-passing distributed systems.
- SW: Understand the various methods for SW fault tolerance. NVP, recovery blocks, run-time checks, problem of predicate detection.

Indicative reading list

Please see Talis Aspire link for most up to date list.

[View reading list on Talis Aspire](#)

Research element

Students are required to based on their project on a scientific research paper. Students will position their project in the group report by incorporating a literature review.

Subject specific skills

Application and systems programming.

Software development processes.

Technical reporting.

Research communication.

Systems analysis and design.

Transferable skills

Technical - Expertise in the analysis and design, operation of dependable computer systems. An understanding of the hardware and software mechanisms that facility the development of dependable computer systems, including the ability to implement these mechanisms.

Communication - Lecture listening. Technical report writing. Technical document comprehension and analysis. Documenting software solutions. Research paper reading. Presentation skills.

Critical Thinking - Systems analysis and technical problem solving. Quantitative performance analysis based. Research project / paper critique.

Multitasking - Management of competing deadlines and priorities. Management of parallel project activities.

Teamwork - Working as part of a technical team in contributing to the development and documentation of a solution.

Creativity - Developing an original solution to a research-based problem.

Leadership - Combining teamwork, critical thinking and technical understanding in the development of a software solution.

Study

Study time

Type	Required
Lectures	20 sessions of 1 hour (13%)
Private study	130 hours (87%)
Total	150 hours

Private study description

Background reading:

N. Lynch, Distributed Algorithms (1st Edition), Morgan Kaufmann, April 1996.

Coursework-related activities:

Reading, programming, systems design, team meetings and project management.

Revision:

Dependability Concepts: Fault, error, failure, fault transformation process. Implications of coverage on dependability, specifications, methods to achieve dependability.

Software: Understand the various methods for SW fault tolerance. NVP, recovery blocks, run-time checks, problem of predicate detection.

Middleware: Protocols for synchronous distributed systems, including leader election, consensus, clock synchronisation, Byzantine agreement and FDIR.

Hardware: Design and analysis of dependable hardware.

Synchronous and asynchronous systems: Protocols and abstractions for asynchronous systems, including logical and vector clocks, broadcast (best-effort, unordered reliable, ordered reliable), failure detectors, global predicate detection in fault-free and faulty systems

Costs

No further costs have been identified for this module.

Assessment

You do not need to pass all assessment components to pass the module.

Students can register for this module without taking any assessment.

Assessment group D1

	Weighting	Study time
Group project	30%	
Having determined a mark for the group submission, credit will be split between group members according to the information you provide on a contribution form.		
Online Examination	70%	
2 hour examination in final chronological term.		
~Platforms - AEP		

-
- Students may use a calculator

Assessment group R

	Weighting	Study time
Online Examination - Resit	100%	
CS347 resit exam		
~Platforms - AEP		

-
- Online examination: No Answerbook required

Feedback on assessment

Written feedback on coursework

Verbal feedback in lectures

[Past exam papers for CS347](#)

Availability

Courses

This module is Optional for:

- UCSA-G4G1 Undergraduate Discrete Mathematics
 - Year 3 of G4G1 Discrete Mathematics
 - Year 3 of G4G1 Discrete Mathematics
- Year 3 of UCSA-G4G3 Undergraduate Discrete Mathematics

This module is Option list A for:

- Year 4 of UCSA-G504 MEng Computer Science (with intercalated year)
- UCSA-G500 Undergraduate Computer Science
 - Year 3 of G500 Computer Science
 - Year 3 of G500 Computer Science
- UCSA-G502 Undergraduate Computer Science (with Intercalated Year)
 - Year 4 of G502 Computer Science with Intercalated Year
 - Year 4 of G502 Computer Science with Intercalated Year
- UCSA-G503 Undergraduate Computer Science MEng
 - Year 3 of G500 Computer Science
 - Year 3 of G503 Computer Science MEng
 - Year 3 of G503 Computer Science MEng

This module is Option list B for:

- USTA-G302 Undergraduate Data Science
 - Year 3 of G302 Data Science
 - Year 3 of G302 Data Science
- Year 3 of USTA-G304 Undergraduate Data Science (MSci)
- Year 4 of USTA-G303 Undergraduate Data Science (with Intercalated Year)