

# CS412-15 Formal Systems Development

**22/23**

**Department**

Computer Science

**Level**

Undergraduate Level 4

**Module leader**

Jane Sinclair

**Credit value**

15

**Module duration**

10 weeks

**Assessment**

Multiple

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

This module uses formal techniques to explore the specification, design and implementation of computer systems. It will cover more advanced features of specification and introduce data and functional refinement, considering how a system design may be modified in a variety of practical ways. Refining the specification to code is an important step in the software life-cycle. This module presents both the theoretical basis for this and practical means of refinement from system specification to program code.

### Module aims

To study methods for designing, analysing and programming complex computing 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.

Revision of sets, relations, functions and sequences.

Revision of basic specification skills.

Introduction to development tool:

- Basic syntax;
- Introduction to the AMN approach;
- Examples and use of tool;  
Case studies;  
Data refinement:
- Refinement;
- Relations and nondeterminism;
- Simulations.  
Functional Refinement;  
Review of refinement concepts using the Guarded Command Language;  
Implementation through refinement;  
Throughout the course there will be an emphasis on tool support;  
We would also hope to include a speaker/s from industry in the course teaching (perhaps for the case studies).

## Learning outcomes

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

- Write formal specifications in one or more chosen notations.
- Understand the principles of formal verification and weakest precondition and the rules for specification consistency.
- Work with online tools to develop specifications; to generate and prove obligations for machines consistency and correctness of refinements; to refine specifications and develop code.
- Understand the principles of refinement and the formal rules that apply. Be able to find loop variants and invariants and to verify loops.

## Indicative reading list

[Reading lists can be found in Talis](#)

[Specific reading list for the module](#)

## Subject specific skills

- Logical reasoning;
- Problem solving;
- Task analysis and decomposition;

## Transferable skills

- Problem solving;
- Team work;

- Communication;
- Logical reasoning;
- Time management;
- Self-directed learning;

---

## Study

### Study time

Type	Required
Lectures	20 sessions of 1 hour (13%)
Supervised practical classes	10 sessions of 1 hour (7%)
Private study	120 hours (80%)
Total	150 hours

### Private study description

Background reading;  
 Completion of exercise sheets;  
 Revision;

### 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 D3

	Weighting	Study time	Eligible for self-certification
Practical assignment 1 Assignment 1: practical coursework task	20%		Yes (extension)
Practical assignment 2 Assignment 2: practical coursework task	20%		Yes (extension)

	<b>Weighting</b>	<b>Study time</b>	<b>Eligible for self-certification</b>
Centrally-timetabled examination (On-campus) 2 hour exam	60%		No

---

- Answerbook Pink (12 page)

## **Assessment group R1**

	<b>Weighting</b>	<b>Study time</b>	<b>Eligible for self-certification</b>
In-person Examination - Resit CS412 Examination resit	100%		No

---

- Answerbook Pink (12 page)
- Students may use a calculator

## **Feedback on assessment**

Individual written feedback on each assignment.

[Past exam papers for CS412](#)

---

## **Availability**

### **Pre-requisites**

MEng students must have studied the material in CS130, CS131, CS262

### **Courses**

This module is Optional for:

- Year 5 of UCSA-G504 MEng Computer Science (with intercalated year)
- Year 1 of TCSA-G5PD Postgraduate Taught Computer Science
- Year 4 of UCSA-G503 Undergraduate Computer Science MEng

This module is Option list B for:

- Year 4 of UCSA-G4G3 Undergraduate Discrete Mathematics