CS141-15 Functional Programming

24/25

Department
Computer Science
Level
Undergraduate Level 1
Module leader
Alexander Dixon
Credit value
15
Module duration
10 weeks
Assessment
Multiple
Study location
University of Warwick main campus, Coventry

Description

Introductory description

The principal aim of this module is to introduce students to the functional programming paradigm.

Module aims

Students should be able to understand the differences between imperative and functional programming, apply functional programming techniques, and write programs in Haskell.

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.

- Differences between imperative and functional programming.
- Functional programming basics: expressions and reduction.
- Types, including parametric polymorphism.
- Ad-hoc polymorphism via type classes.
- Recursive and higher-order functions.
- Algebraic data types.
- Strict vs Lazy evaluation.
- Inductive construction.

- Functors, Applicatives, and Monads.
- Practical Haskell programming.

Learning outcomes

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

- Understand the differences between programming paradigms as well as their strengths and weaknesses in order to identify suitable programming languages for a particular task's needs.
- Apply key techniques of the functional programming paradigm to solve programming problems.
- Identify and exploit patterns in programs to design and implement programming abstractions.
- Prove simple properties about functionally pure computer programs.
- Use different evaluation strategies to evaluate programs.

Indicative reading list

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

View reading list on Talis Aspire

Subject specific skills

Understanding of Functional Programming as a programming paradigm, including intermediate knowledge of programming abstractions and formal reasoning. See syllabus for details.

Transferable skills

Technical skills.
Critical thinking.
Multitasking.

Study

Study time

Type	Required
Lectures	30 sessions of 1 hour (20%)
Practical classes	10 sessions of 1 hour (7%)
Private study	110 hours (73%)
Total	150 hours

Private study description

Background reading of recommended texts.
Work on unsupervised practical assignments.
Exam 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 D4

	Weighting	Study time
Coursework 1	15%	
This assessment is eligible for self-certification (extension).		
Coursework 2	25%	
Coursework 2. This assignment is worth more than 3 CATS and is not, therefore, eligible for self-certification.		
In-person Examination	60%	
CS141 exam		

- Answerbook Pink (12 page)

Assessment group R2

	Weighting	Study time
In-person Examination - Resit	100%	
CS141 resit examination		

- Answerbook Pink (12 page)

Feedback on assessment

Individual feedback on coursework via Tabula

Past exam papers for CS141

Availability

Courses

This module is Optional for:

- UCSA-G500 Undergraduate Computer Science
  - Year 1 of G500 Computer Science
  - Year 1 of G500 Computer Science
- UCSA-G503 Undergraduate Computer Science MEng
  - Year 1 of G500 Computer Science
  - Year 1 of G503 Computer Science MEng
  - Year 1 of G503 Computer Science MEng
- Year 1 of UCSA-I1N1 Undergraduate Computer Science with Business Studies
- Year 1 of UCSA-G406 Undergraduate Computer Systems Engineering
- Year 1 of UCSA-G408 Undergraduate Computer Systems Engineering

This module is Option list B for:

- UCSA-G4G1 Undergraduate Discrete Mathematics
  - Year 1 of G4G1 Discrete Mathematics
  - Year 1 of G4G1 Discrete Mathematics
- Year 1 of UCSA-G4G3 Undergraduate Discrete Mathematics