

# CS126-15 Design of Information Structures

**24/25**

**Department**

Computer Science

**Level**

Undergraduate Level 1

**Module leader**

Richard Kirk

**Credit value**

15

**Module duration**

10 weeks

**Assessment**

Multiple

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

CS126 is all about data structures and how to program them.

We are interested in: basic data structures; how to mathematically analyse them; how we can program those data structures; how we can represent them efficiently; how we can reason about them (in a formal manner).

We are also interested in common algorithms that use data structures, including: searching for data; sorting data.

### Module aims

The module aims for students to:

- gain familiarity with the specification, implementation and use of some standard abstract data types (ADTs) such as linked-lists, stacks, queues, graphs etc.
- learn some standard algorithms for common tasks (such as searching and sorting) and some elementary methods of measuring the complexity, and of showing the correctness, of algorithms;

- learn how to program with non-standard ADTs using an object-oriented language.

## 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.

- Types and their properties: simple types in programming languages; relationship between familiar mathematical and program objects of given type. Using predicate logic to state properties of types and their operations in terms of pre- and post-conditions.
- Abstract data types: specification of familiar abstract objects (eg complex numbers, sets, sequences, matrices) and their operations, comparison with their implementation using a typical programming language. Specification and implementation of some important standard types (eg strings, stacks and queues).
- Algorithms: relationship between data structures and algorithms; some standard algorithms for searching, sorting and pattern matching. Elementary analysis of complexity. Reasoning about the correctness of the implementation of simple algorithms.

## Learning outcomes

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

- After completing CS126 Design of Information Structures, a student should be familiar with a range of standard ADTs and how they can be used to accomplish common programming tasks.
- After completing CS126 Design of Information Structures, a student should be able to assess the complexity and correctness of simple algorithms, and choose appropriate algorithms for simple tasks.
- After completing CS126 Design of Information Structures, a student should have practical experience of designing user-defined ADTs, and associated algorithms, for a non-standard application.

## Indicative reading list

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

[View reading list on Talis Aspire](#)

## Subject specific skills

- Specifying abstract data types and implementing them in an object-oriented programming language
- Estimating the asymptotic running time of simple algorithms
- Using basic data structures to implement efficient algorithms

## Transferable skills

- Creative problem solving
- 

## Study

### Study time

Type	Required
Lectures	30 sessions of 1 hour (20%)
Practical classes	8 sessions of 2 hours (11%)
Private study	104 hours (69%)
Total	150 hours

### Private study description

- Background reading.
- Creative problem solving, either individually or in groups.
- Programming for the lab assignments and coursework.
- 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 C4

	Weighting	Study time	Eligible for self-certification
Lab session 1	2%		Yes (waive)
Marked Laboratory session Week 3			
Lab session 2	2%		Yes (waive)
Marked Laboratory session Week 4			
Lab session 3	2%		Yes (waive)

	<b>Weighting</b>	<b>Study time</b>	<b>Eligible for self-certification</b>
Marked Laboratory session Week 5			
Lab session 4	2%		Yes (waive)
Marked Laboratory session Week 6			
Lab session 5	2%		Yes (waive)
Marked Laboratory sessions Week 7			
Programming assignment	40%		No
Programming Assignment..			
In-person Examination	50%		No
Examination			

- 
- Answerbook Pink (12 page)

## Assessment group R2

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

- 
- Answerbook Pink (12 page)

## Feedback on assessment

Mark and written feedback for coursework returned via Tabula.

[Past exam papers for CS126](#)

---

## Availability

### Pre-requisites

Students must have studied the material in CS118 or MA117

## Courses

This module is Core for:

- Year 1 of UCSA-G500 Undergraduate Computer Science
- UCSA-G503 Undergraduate Computer Science MEng
  - Year 1 of G500 Computer Science
  - 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
- Year 1 of USTA-G302 Undergraduate Data Science
- Year 1 of USTA-G304 Undergraduate Data Science (MSci)
- Year 1 of UCSA-G4G1 Undergraduate Discrete Mathematics
- Year 1 of UCSA-G4G3 Undergraduate Discrete Mathematics

This module is Optional for:

- Year 1 of USTA-G1G3 Undergraduate Mathematics and Statistics (BSc MMathStat)
- Year 1 of USTA-GG14 Undergraduate Mathematics and Statistics (BSc)

This module is Option list B for:

- Year 1 of UMAA-GV18 Undergraduate Mathematics and Philosophy with Intercalated Year