

CA215 Languages and Computability

	Computability Theory	Programming
<i>Lecturer</i>	Andy Way	Mary Hearne
<i>Office</i>	L245	L215
<i>Phone</i>	5644	8486
<i>Email</i>	away@computing.dcu.ie	mhearne@...

Course webpage:

<http://www.computing.dcu.ie/~away/CA215>

Recommended Texts: *"Elements of the Theory of Computation"*, H.R. Lewis & C.H. Papadimitriou, Prentice Hall, 1998 (2nd edition), [511.3/LEW]
Haskell: The Craft of Functional Programming, S. Thompson, Addison Wesley, 1999 (2nd edition), [005.133/THO]

Assessment

Component	Weight	When	What
Programming exam	25%	Week 10	Lab Test
End-of-year exam	75%	January	10 questions

Schedule

Lectures

Day	Time	Location	Type	Lecturer
Tues	11:00	QG21	Computability Theory	AW
Weds	11:00	XG22	Functional Programming	MH

Labs

Day	Time	Location	Type
Tues	16:00–18:00	L114/201	Functional Programming

Person responsible for Labs is Grzegorz Chrupala:
gchrupala@computing.dcu.ie

Tutorials

Day	Time	Location	Type
Mon	16:00	C166	Computability Theory
Mon	16:00	Q119	Computability Theory

Person responsible for Tutorials is Natalie Schluter:
nschluter@computing.dcu.ie

What is this Course about?

We will be answering the question: *What are the limits of what can be solved by computer?*

- Computers only have a finite amount of RAM, so there are problems which are too large for it to solve.
- Therefore we abstract away from these limitations and consider what can be solved by the 'most general' computing device, one which is not limited in its memory capacity - but is still required to produce an answer in a finite amount of time.

Computability theory is concerned with exploring the limitations of such idealized computing devices.

- Computation can be regarded as a process of *language recognition*.
- We will study a number of languages of increasing descriptive power.
- We will also study a number of abstract machines of increasing power, which can be used to recognise each of these languages.

- Some of the machines which we will study are as powerful as any real computer, while others are less powerful.
- We will see that these less powerful machines are nevertheless quite useful, and ideally suited to certain tasks.

Course Outline

- Section A: Functional Programming
- Section B: Computability Theory
 1. Mathematical Prerequisites
 2. Regular Languages and Finite Automata
 3. Context-Free Languages and Pushdown Automata
 4. Context-Sensitive Languages and Linear-Bounded Automata
 5. Unrestricted Languages, Turing Machines and the λ -Calculus
 6. Unsolvable Problems

“Computer Science is no more about computers than astronomy is about telescopes.” - Edsger W. Dijkstra