

## Data Structures and Algorithms Introduction

Phil Trinder  
email: P.W.Trinder@hw.ac.uk  
room: G.52

1

## About the Course

- Focuses on useful **data structures** and **algorithms** for range of practical applications.
- Mainly independent of programming language (but Java examples given).
- Specific topics include:
- String processing algorithms
  - Compression
  - Cryptography
- Graph data structures and algorithms.
- Algorithm Design Techniques
- Complexity Analysis

2

## Module support

Available via [www.macs.hw.ac.uk/~trinder](http://www.macs.hw.ac.uk/~trinder)  
includes

- Course outline
- Information on books, including access to software
- Lecture Notes
- Tutorial sheets
- Assessment details including coursework handout, submission instructions

3

## Survival Guide

- Basics
  - Attend & participate in lectures, labs & tutorials
  - Review notes, do exercises
  - Identify problem areas
  - Get help: in labs, tutorials, friends
- Approach
  - **Think!** Understand the problem
  - Construct a pencil and paper solution
  - Only produce code when you know what's going on
- Important Metrics
  - A 'working' program may not be a good program
  - **Correct**
  - **Clarity:** good structure, simplicity, comments
  - **Efficient** in (execution) time and space (memory usage)

4

## Core Technical Competence for Employability

- A major Scottish employer lists, with 6 other competences:
- **Algorithms** Good understanding of big O, should know common algorithms for searching and sorting and their characteristics e.g. binary search, quicksort.
- **Data Structures** know about hashables (including hashing functions), stacks, linked lists, double-linked lists and operations on them. Bonus points for knowledge of more unusual structures.