Data Structures and Algorithms Introduction

Hans-Wolfgang Loidl
email: H.W.Loidl@hw.ac.uk
url: `http://www.macs.hw.ac.uk/~hwloidl`
room: G.48

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

Module support

Available via `http://www.macs.hw.ac.uk/~hwloidl/Courses/F28DA` includes

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

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)

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 hashtables (including hashing functions), stacks, linked lists, double-linked lists and operations on them. Bonus points for knowledge of more unusual structures.

Knowledge Assumed

We assume you know

- basic data structures, such as arrays, bags, lists, stacks, queues, linked lists.
- basic mathematical concepts, such as functions, logarithms, exponents, sets, series, derivatives, limits
- the concept of abstract data types
- basic object oriented design concepts, such as abstraction, encapsulation, modularity

Skills Assumed

We assume you know how to program in Java and that you

- can design, implement, test, debug, and document simple Java programs
- can read and understand moderate size Java programs
- can write and understand simple recursive code

This is **not** a course on Java. If you don't know Java, start learning it now, before the programming assignments are due!

Knowledge and Skills Gained

In this course you will learn

- how to design moderately complex data structures, fitting a particular problem domain,
- how to develop efficient algorithms, solving the given problem,
- how to analyse the algorithm and predict its performance

Reading List

References

[Goo11] Michael T. Goodrich and Roberto Tamassia, *"Data Structures and Algorithms in Java"*, 5th edition, John Wiley & Sons, 2011. ISBN: 978-0-470-39880-7

[Prei00] Bruno R. Preiss, *"Data Structures and Algorithms with Object Oriented Design Patterns in Java*, John Wiley & Sons, 2000.

[ArGo96] Arnold and Gosling, *"The Java Programming Language*, Addison-Wesley, 1996.

[ArWe99] Arnow and Weiss, *"Java, an Object Oriented Approach"*, Addison-Wesley, 1999.