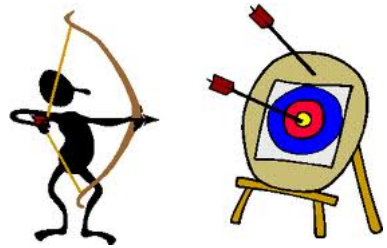
Constructing Software Systems

Essential Computing II

by
Keld Helsgaun

Agenda

- **Purpose of the course**
- **Algorithms and data structures**
- **Prerequisites**
- **Schedule of lectures**
- **Exam**



Purpose

The purpose of this course is to provide a practical introduction to **algorithms** and **data structures** from the viewpoint of abstract thinking and problem solving.

The primary focus is on **problem-solving techniques** that allow the construction of sophisticated time-efficient programs.

What is an algorithm?

An **algorithm** is a method for solving a problem

Notice that it is not required that an algorithm is executed on a computer.

This course deals primarily with computer algorithms.

Three important areas of interest:

- **Design**
- **Analysis**
- **Verification**

Design: How do we construct an algorithm?

Analysis: What are the resource demands of an algorithm?

Verification: Can we guarantee the correctness of an algorithm?

Why study algorithms?

(1) To make a qualified choice among existing algorithms.

(2) To adapt an existing algorithm for a given purpose.

(3) To develop new algorithms.

Donald Knuth:
My favorite way to describe computer science is to say that it is the study of algorithms.

The data concept

Data:

A *formalized representation* of facts or concepts suitable for communication, interpretation, or processing by people or automated means. Data on its own carries no meaning.

Information:

The meaning that a human assigns to data by means of known conventions.

Datalogy:

The science of data and data processes.

What is a data structure?

A **data structure** is a particular way of storing and organizing data in a computer so that it can be used efficiently

Examples:

stack, queue, linked list, tree, hash table, priority queue

Implementation in Java:

by simple variables, references, arrays, and classes

Prerequisites

Students should have knowledge of either an object-oriented or procedural language. Knowledge of basic features, including primitive data types, operators, control structures, functions (methods), and input and output is assumed.

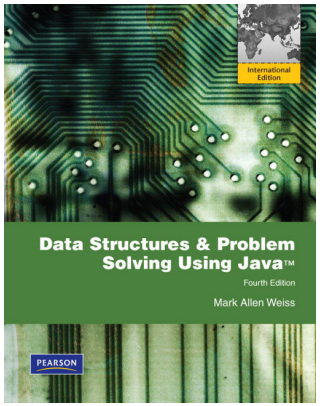
Knowledge of Java is not assumed.

Schedule of lectures

ECII

(1) Preliminaries I — Java in two weeks
(2) Preliminaries II
(3) Algorithms I
(4) Algorithms II
(5) Algorithms III
(6) Implementations I
(7) Implementations II
(8) Implementations III
(9) Thread programming
(10) Applications I — Games, parsing
(11) Applications II — File compression, simulation
(12) Applications III — Graphs

Text book

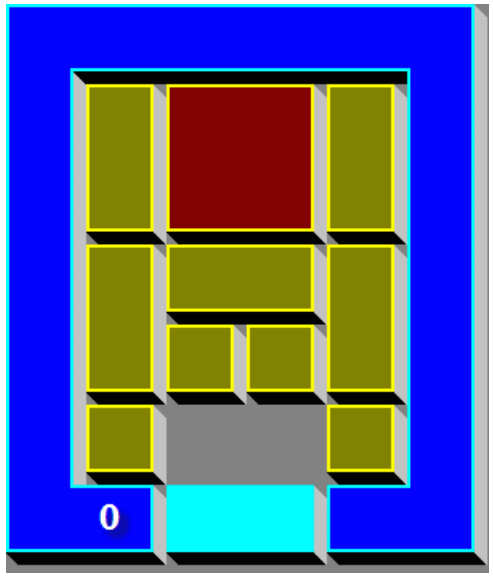


Data Structures & Problem Solving Using Java
Mark Allen Weiss,
Pearson, 4th edition, 2010.

Advantages:

- good explanations in easy-to-read English
- emphasis on data abstraction (hiding away implementation details)
- good examples
- algorithms are expressed in executable code

Exam



15 minutes oral examination based on a practical assignment given during the course.

The exam will focus on the solution of the assignment, and a randomly drawn exam question.

Web page

(available via www.ruc.dk/~keld)

Roskilde Universitetscenter ▲ Datalogi ▲

Constructing Software Systems

Spring 2014

by

Keld Helsgaun

Contents

The course provides knowledge about fundamental algorithms and data structures in computer science. The main objective of this course is to provide students with means to design and reason about algorithms, understand the strengths and weaknesses of known algorithms, and their suitability in particular contexts. The topics covered include data abstraction, generic components, algorithm analysis, recursion, algorithm design, searching, sorting, randomization, simulation, and graphs. In more detail, the contents of the course may be described as follows:

Plan 0

January 29 – February 13

- Read Chapter 1 and 2 in the textbook

Time and place

Lectures and **Exercises**: Thursdays 8^{30}-12^{45}.

Room 43-2.29. First lecture: Thursday, February 13.


That's all Folks!