Abstract Data Types (ADT's)

1. ADT a model with a collection of operations defined on that model

Example 1: a set with operations for adding an element, testing for membership, union, intersection, diffrence and so on

Example 2: a person, with operations for birth, marriage, death, aging, changing address and/or job, retiring, etc

Example 3: a graph, with operations for getting uncoloured nodes, incident edges, adjacent nodes, deleting an edge, adding an edge, and so on

2. We may implement the ADT in many different ways, but its behaviour/function should be independent of that implementation (think of binary versus decimal numbers, addition is the same)

3. We will look at some ADT's in detail, show different ways of implementing them (usually 3 different ways) and discuss advantages and disadvantages of each.

4. We will look at lists, dlists, stacks, queues, trees, binary trees, sets, hashing, directed graphs, and graphs

Abstract Data Types

DATA TYPE: of a variable is set of values it may assume such as real, integer, boolean, character, string, ...

ABSTRACT DATA TYPE: mathematical model, together with operations and is generally represented in terms of data types within the language used

DATA STRUCTURE: a collection of variables, structured in some way ... and we use this to implement the model.
In scheme we will be using the record.

Coding Conventions etc
/home/s7/pat/scheme/pub/world.scm

0. Alternative syntax to (define f (lambda (x) ...))
and why I do it that way, and is it different anyway?

1. ARRAYS: n dimensional arrays, what is an n dimensional array, how does it work, why do I want it?
/home/s7/pat/pub/arrays.scm

2. Iteration, via do
/home/s7/pat/scheme/pub/do.scm

3. IO, reading and writing to a file
/home/s7/pat/scheme/pub/io.scm

4. records
example in /home/s7/pat/scheme/pub/record.scm
manual in /home/s7/pat/scheme/pub/record.text

YOU NEED TO BE COMPETENT IN THIS TO SURVIVE ....