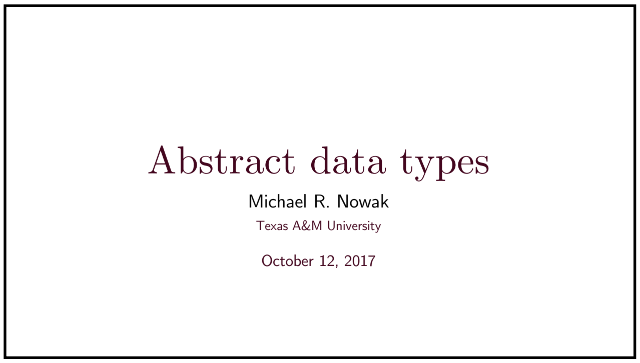
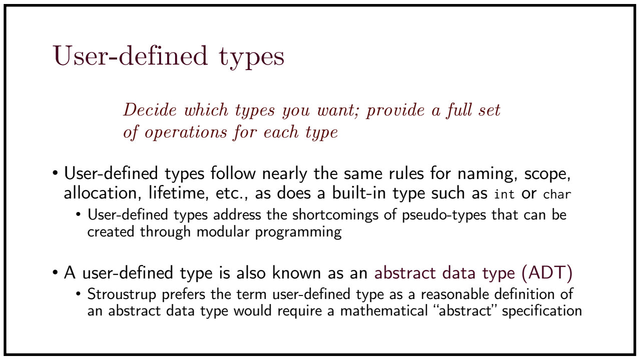


Abstract data types

Michael R. Nowak

Texas A&M University

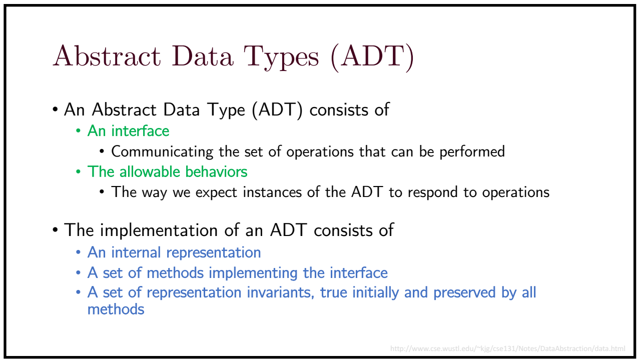
October 12, 2017



User-defined types

Decide which types you want; provide a full set of operations for each type

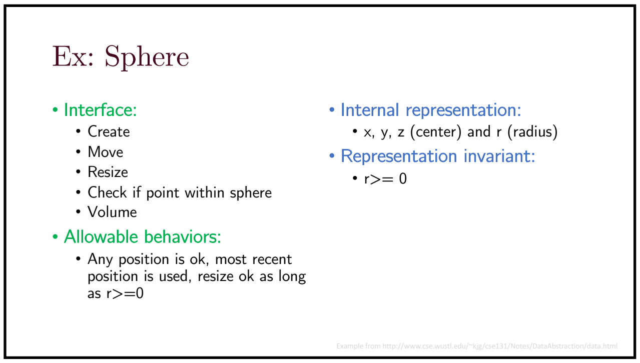
- User-defined types follow nearly the same rules for naming, scope, allocation, lifetime, etc., as does a built-in type such as `int` or `char`
  - User-defined types address the shortcomings of pseudo-types that can be created through modular programming
- A user-defined type is also known as an abstract data type (ADT)
  - Stroustrup prefers the term user-defined type as a reasonable definition of an abstract data type would require a mathematical "abstract" specification



Abstract Data Types (ADT)

- An Abstract Data Type (ADT) consists of
  - An interface
    - Communicating the set of operations that can be performed
  - The allowable behaviors
    - The way we expect instances of the ADT to respond to operations
- The implementation of an ADT consists of
  - An internal representation
  - A set of methods implementing the interface
  - A set of representation invariants, true initially and preserved by all methods

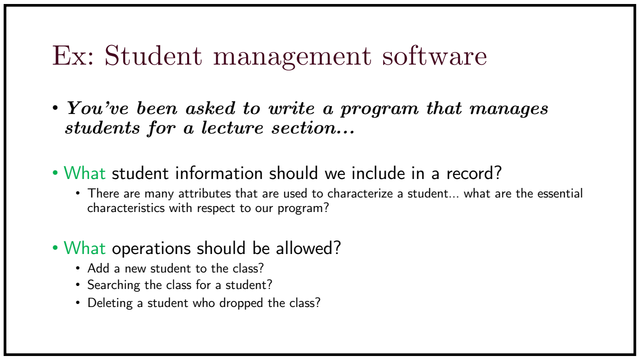
http://www.cse.wustl.edu/~kjg/cse131/Notes/DataAbstraction/data.html



Ex: Sphere

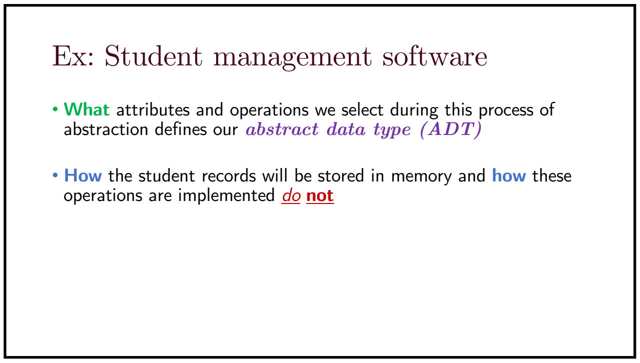
- Interface:
  - Create
  - Move
  - Resize
  - Check if point within sphere
  - Volume
- Allowable behaviors:
  - Any position is ok, most recent position is used, resize ok as long as r>=0
- Internal representation:
  - x, y, z (center) and r (radius)
- Representation invariant:
  - r>= 0

Example from http://www.cse.wustl.edu/~kjg/cse131/Notes/DataAbstraction/data.html



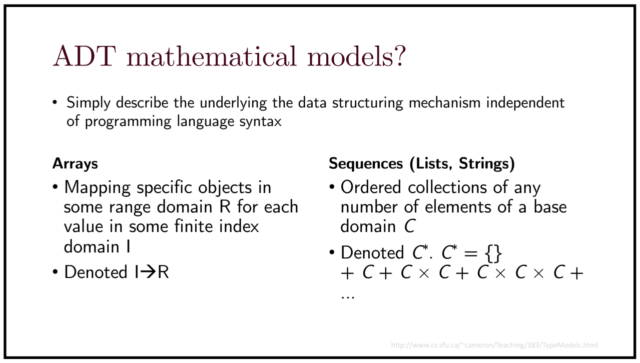
Ex: Student management software

- ***You've been asked to write a program that manages students for a lecture section…***
- What student information should we include in a record?
  - There are many attributes that are used to characterize a student… what are the essential characteristics with respect to our program?
- What operations should be allowed?
  - Add a new student to the class?
  - Searching the class for a student?
  - Deleting a student who dropped the class?



Ex: Student management software

- **What** attributes and operations we select during this process of abstraction defines our ***abstract data type (ADT)***
- **How** the student records will be stored in memory and **how** these operations are implemented *do* **not**



ADT mathematical models?

- Simply describe the underlying the data structuring mechanism independent of programming language syntax

Arrays

- Mapping specific objects in some range domain R for each value in some finite index domain I
- Denoted I→R

Sequences (Lists, Strings)

- Ordered collections of any number of elements of a base domain C
- Denoted C^*. $C^* = \{\} + C + C \times C + C \times C \times C + \ldots$

http://www.cs.sfu.ca/~cameron/Teaching/383/TypeModels.html