CSC151 2010S, Class 01: An Introduction to the Course

- Hi, I’m Sam. Your class Mentor is Lea.

http://www.cs.grinnell.edu/~rebelsky/Courses/CSC207/2013F
http://www.cs.grinnell.edu/~rebelsky/Courses/CSC207/2013F/eboards/01.md

Overview

- Admin.
- About the Course.
- ADTs and Data Structures.
- An Exercise: Designing a List ADT.

Admin

- Welcome to CSC 207. I look forward to helping you learn.
- Since this is a new course, I had lots of work to do on the course web. The web is incomplete. Sorry.
- I don’t believe in going over class policies on the first day. Please read the course web and ask me questions via irc, email, or in the next class.
- MathLAN has a brand new desktop. Blame Adam and Jordan.

About the Course

- Topics
  - Problem solving
  - Algorithms
  - Analysis - Understanding our solutions
  - Data Structures and Abstract Data Types
  - Software Design, particularly the object-oriented paradigm
  - Program
  - The Language Java, even though it’s not our focus

- Learning techniques
  - Labs (Collaborative, constructivist learning situations) ADTs and Data Structures
  - AND MORE

- Skills
Teamwork
Um or Uh
AND MORE

● Outcomes
  ○ Generalize
  ○ AND MORE

● We’ll be doing the topics in more depth

Algorithms
  ● Classic algorithms
  ● Formal analysis of running time and space usage
  ● Look at patterns of algorithm design
    ○ divide-and-conquer
    ○ greed
    ○ dynamic programming
  ● Write algorithms for problems
  ● Implement them in Java

Data Structures and ADTs
  ● Classics
  ● Analyze (typically in terms of the algorithms that use them)
    ○ You know about linked lists and arrays/vectors
  ● Design and implement DSs
  ● Generalize to ADTs (or start general)

Program Design
  ● Larger programs
  ● Using OOD principles
    ○ Encapsulation - Group things together that belong together and protect them from the outside world
    ○ Inheritance - Aristotle was right. Things are hierarchical. Easy ways to build new code from old by extending it.
    ○ Polymorphism

Course Themes
  ● Android development
  ● Computing for Social Good
  ● Trying to limit impact on core topics
  ● Probably enhances other topics
Sam’s Steps in ADT/DS Design

- ADT: Abstract, the *what*
- DS: Concrete, the *how*

Philosophy: What is the main driving approach?
- For a vector, "values indexed by sequential integers"

Problem Domains: How would we use that organization
- Registrar, Mailboxes, etc.

Procedures: What are the ways in which someone will use this ADT?
- Add or set a value at a particular index `add(vector,index,thing)` `set(vector,index,thing)`
- Remove a value at a particular index shifting everything that follows that index? just leaving it empty?
- `get(vector,index)`
- `containsValue(vector,index)` or `isEmpty(vector,index)`
- Create a new vector
- Delete a vector

Data Structures

Decide on a layout of memory

Design strategy for implementing procedures

Implement

Analyze

Copyright (c) 2013 Samuel A. Rebelsky.

This work is licensed under a [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/) To view a copy of this license, visit [http://creativecommons.org/licenses/by/3.0/] or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.