Announcements

bit-bug for problems with submission, your code, the any of our software.

lab assistants needed. Consider volunteering to be b assistant for CS 10, self-paced courses, CS 61A, or semester.

Lecture #40: Course Summary

language: Java
lysis
f data structure: Java library structure

om numbers

plementation topics

Programming-Language Topics

d programming: organizing around data types
nted programming:
s. static type
ce
terface vs. implementation
ramming (the <···> stuff).
el: containers, pointers, arrays
es
and semantics
xtent
oms, patterns:
sed as functions (e.g., Comparator)
plementations (e.g., AbstractList)

., sublists)

alysis and Algorithmic Techniques

analysis
), $\Theta(\cdot)$ notations
average case.
ime
and dynamic programming.

jor Categories of Data Structure

terface and its subtypes
e and its subtypes
eton implementations of collections, lists, maps (`AbstractList`,

ncrete collection and map classes in Java library

Sequences

double link manipulations

rrays
es, deques
fering
costs of basic operations

Trees

s: search, representing hierarchical structures
ions: insertion, deletion
sals
trees

Sorting

ing
rt
rting

d selection
sort

f various algorithms, when to use them?

Searching

s, range searching
onal searches: quad trees.

es and heaps
es
ng by rotation (red-black trees)
y construction (B-trees)
tic balance (skip lists)

s, trade-offs

Graph structures

represented by graphs
sal: the generic traversal template
traversal, breadth-first traversal
ort
ths
ning trees, union-find structures
agement as a graph problem.

Random numbers

eudo-random sequence
uential and additive generators
tributions:
the range
rm distributions
ndom selection

Version Control

?
ts behind our particular system:
opy vs. repository copy
g changes
and merging changes.

Debugging

gers can do
o pin down bugs
me debugger (Eclipse, gjdb, various Windows/Sun prod-
).
what it means, how to use it.
nics.

A Case Study

t version-control system as an example of a design using
s from this course.

and tree structures represented with files as vertices
file names), rather than machine addresses, as pointers.

ing to create unique (or very, very likely to be unique)
abilistic data structure.

uses various kinds of map to facilitate conversion to
npressed form, including arrays, tries, and hash tables

e in Huffman coding.

Assorted Side Trips

essing.

agement and garbage collection.

What's After the Lower Division?

Principles & Techniques of Data Science (Gonzalez, Ad-

Interface Design (Hartmann)

uter Security (Wagner, Popa)

rating Systems and System Programming (Kubiatowicz)

gramming Languages and Compilers (Hilfinger, Sen)]

o. to the Internet: Architecture and Protocols (Rat-

cient Algorithms and Intractable Problems (Chiesa, Nel-

binatorics and Discrete Probability]

ep Neural Networks (Canny)

hics (Ng)

bases (Hug, Ball)

ficial Intelligence (Rao, Lambert)

What's After the Lower Division? (II)

hine Learning (Shewchuk)

ntum Information Science and Technology

rted Special Topics: Image Manipulation, Computer Vi-
putational Photography (Efros)

al Implications of Computer Technology (Hug, Ball)

puter Architecture (Asanovic)

tware Engineering]

raduate courses: including advanced versions of 152,
0, 184, 186, 189; plus Cryptography, VLSI design and
topics.

se, EE courses!

rtunities for participating in research and independent

What's After the Lower Division? (III)

CS are just two of over 150 subjects!

offer more specific skills and exposure to real prob-

think that CS is a creative activity that (to the true
t to fun!