Announcements

on December 5, 6, and 7 will be organized as follows:
l work on an exam-like set of exercises covering linked
, queues, binary trees, binary search trees. Solutions
ughly reviewed. 1 bonus point (out of 200) for complet-
cises.

it-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.

Contest: Visit my web page for information about the
amming contest, which we hold each fall. There are
ions of programming problems you can try your hand on.

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

nalysis

), $\Theta(\cdot)$ 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

d double link manipulations

rrays

es, deques

fering

costs of basic operations

Trees

s: search, representing hierarchical structures

ions: insertion, deletion

sals

g trees

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

Sorting

ing

rt

rting

d selection

sort

f various algorithms, when to use them?

Random numbers

s

eudo-random sequence

uential and additive generators

tributions:

the range

rm distributions

ndom selection

Graph structures

represented by graphs

rsal: the generic traversal template

traversal, breadth-first traversal

ort

ths

ning trees, union-find structures

agement as a graph problem.

Debugging

gers can do

o pin down bugs

me debugger (Eclipse, gjdb, various Windows/Sun prod-

what it means, how to use it.

nics.

Version Control

r?

ts behind our particular system:

copy vs. repository copy

g changes

and merging changes.

A Case Study

t version-control system as an example of a design using
; 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.

Vhat's After the Lower Division?

Interface Design (Hartmann)

uter Security (Popa)

ating Systems and System Programming (Joseph, Ragan-

ramming Languages and Compilers (Hilfinger)

:ient Algorithms and Intractable Problems (Chiesa, Vazi-

oinatorics and Discrete Probability (Friedman)

hics (Ng)

bases

icial Intelligence (Dragan, Levine)

iine Learning

orted Special Topics: Computational Design and Fabri-
gning, Visualizing and Understanding Deep Neural Net-

at's After the Lower Division? (II)

uter Architecture (Asanovic)

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

se, EE courses!

rtunities for participating in research and independent

t's After the Lower Division? (III)

'S 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!