Course Syllabus
Introduction to Computer Science II (ICS211)
Section 3, Fall Semester 2013

Instructors

Martha E. Crosby, Ph.D.
Office: POST 305D
Office Hours: T 3:00-4:00, W 11:00-12:00
Email address: crosby@hawaii.edu

Anthony Christe
Office: POST 303-5
Office Hours: W 1:30 - 2:30 PM
Email address: achriste@hawaii.edu

Course Description

The prerequisites for Introduction to Computer Science II (ICS211) are Introduction to Computer Science I (ICS111), previous or concurrent Discrete Math (ICS141) or consent. Material in ICS211 concentrates on: the theory of algorithms and their complexity; an introduction to software engineering; an introduction to data structures; and searching and sorting algorithms.

Objectives

The objectives of this course are to increase the student's ability to:

- create projects that involve three classes or more
- understand collection classes and the basics of memory management
- understand the importance of implementing exception handling in their code
- implement the following ADTs and data structures:
  - Stacks
  - Queues
  - linked lists
  - binary trees
  - heaps
  - hash tables
- understand and be able to calculate the Big-O of an algorithm
- programmed at least one sorting algorithm, and
- understand the following sorting algorithms:
  - bubble sort
  - insertion sort
  - selection sort
  - merge sort
  - heap sort
  - quick sort
- understand encapsulation, information hiding and ADT concepts and uses
- understand and use and program tree traversal algorithms
- write algebraic equations in prefix, infix and postfix notation
- use preorder, inorder and postorder traversals
- understand recursion and be able to write simple recursive methods
- understand the mechanism of Huffman encoding

Course Textbook (required)

Data Structures: Abstraction and Design Using Java, Second Edition
Elliot B. Koffman and Paul A. T. Wolfgang,
Wiley 2010. ISBN: 978-0-470-12870-1

Course Policies on Integrity, Promptness

- Plagiarism will not be tolerated. When assignments require team work, all team members must be active and participate equally in the solution to the problem
- Assignments will be due on Fridays at 11:59 HST. Late assignments will not be accepted (you will receive no credit for the assignment).

GitHub Repository for ICS211 section 3 Policies and Resources (https://github.com/anthonyjchriste/ics211f13/wiki#ics-211-lab-links)

- Lab Links (Notes and Assignments)
- Resources (Github and Programming)

Grading

- Homework 45%
- Participation 5%
- Exams 1 (15%), Exam2 (15%), Final Exam (20%)
- Possible Extra Credit (5%)

KOKUA

Any student who feels s/he may need an accommodation based on the impact of a disability is invited to contact the KOKUA Program. We would be happy to work with you, and the KOKUA Program (Office for Students with Disabilities) to ensure reasonable accommodations in the course. KOKUA can be reached at (808) 956-7511 or (808) 956-7612 (voice/text) in room 013 of the Queen Lili'uokalani Center for Student Services.

Class Schedule

Week	Date	Textbook Reference	Topics
1	26 Aug		Introductions, Overview and Class Orientation
	28-Aug	Appendix A	Java review:
2	2-Sep	Labor Day	No class
	4-Sep	Sections 1.1-1.2	Java review: basics; exceptions; variables; arrays; modulo operator; if statements, boolean comparisons, loops: while and for
3	9-Sep	Sections 1.3-1.8, A6, A12	overriding methods, overloading methods, and polymorphism; casting and instanceof/getClass; Abstract classes; class Object; Exception class hierarchy
	11-Sep	Sections 2-2.4	program runtime; algorithm efficiency; Big-O notation; list interface; array lists
4	16-Sep	Sections 2.4-2.5	list interface; array lists; linked lists; nodes; implementation of linked list; invariants (not in text)
	18-Sep	Sections 2.5-2.6	linked lists; nodes; implementation of linked list; invariants; circular linked list; doubly-linked lists; iterators; iterator implementation; the ListIterator interface; the Java foreach statement
5	23-Sep	Sections 2.6-2.9	iterators; iterator implementation; the Java foreach statement
	25-Sep	Sections 2.10-2.11	testing; errors; reasoning about programs
6	30-Sep	Exam review Chapters 1 & 2, Appendix A6, A12.	Java review; Abstract Data Types (ADTs); Interfaces; Class Hierarchy; Abstract Classes; Inheritance; Invariants; Lists; ArrayList; LinkedList; runtime analysis; Iterators; references
	2-Oct	covers all material to date	Exam 1
7	7-Oct	Sections 3.1, 3.3	post-exam review; stacks; stack ADT; method signatures; array stack implementation

	9-Oct	Chapter 3	linked stack implementation; stack applications; infix, prefix, and postfix expressions
8	14-Oct	Chapter 4	queues; queue interface; queue applications; queue implementation: array queue; queue implementation: linked queue
	16-Oct	Sections 4.3-4.5	application of queues and stacks: data structure traversal; double-ended queues; application of queues: simulation of an airline counter; random numbers; recursion; examples of recursion; implementation of recursion using a stack; principles of recursion; binary search; recursion examples
9	21-Oct	Sections 5-5.3	principles of recursion; binary search; more examples of recursion; recursive linked list methods; problem solving with recursion
	23-Oct	Sections 5.2, 5.3-5.5	recursive linked list methods; problem solving with recursion;
10	28-Oct	Sections 6.1 -6.2	Binary Trees; Types of Binary Trees; Binary Search Trees;
	30-Oct	Sections 6.2-6.3	Visualizing Tree Traversals; Traversal of Binary Search Trees and Expression Trees; Implementing Binary Tree Class; Node<E> Class; Binary Tree Class
11	4-Nov	Sections 6.4	Overview of Binary SearchTree; Performance; Interface SearchTree; Binary Tree Class; binary search tree algorithms: add, remove, traverse; Testing Binary Search Tree;
	6-Nov	Sections 6.5	Heaps and Priority Queues; heap insertion and removal; Implementing a Heap; heap storage in arrays; Priority Queues
12	11-Nov Veteran's Day		No class
	13-Nov	Section 6.5	Priority Queues Class; Using a Heap as the Basis of a Priority Queue; Other Methods

13	18-Nov Exam review	stacks; queues; infix, prefix, and postfix expressions; random numbers; recursion; binary search; binary trees; binary search trees; tree traversal; heaps; priority queses; runtime analysis; ability to implement needed methods
	20-Nov  primarily material since exam 1	Exam 2
14	25-Nov Section 6.6	Post-exam review; Huffman coding; Huffman trees; Implementation of Huffman coding; hash tables; hash functions; open addressing; chained hashing
	27-Nov Sections 6.6, 7.3	Implementation of Huffman coding; hash tables; hash functions; open addressing; chained hashing
15	2-Dec Sections 7.3, 7.4	open addressing; chained hashing; applications of hashing; equality and comparisons in Java
	4-Dec Section 8-8.4	sorting; selection sort; bubble sort; insertion sort
16	9-Dec. Sections 8.4-8.9	merge sort; heap sort; quick sort; shell sort; 2-3 trees
	11-Dec course review:	exam 1 material: Java, ADTs, linked lists, run-time big-O, objects, references and pointers, iterators, invariants; exam 2 material: generic types and container classes (including vectors), stacks, queues, recursion binary trees, binary search, binary search trees, tree traversal, heaps, huffman coding, priority queues, hashing, sorting: insertion, selection, bubble, mergesort, heapsort, quicksort

 Final Exam Your choice 4:30pm either on December 16th or 18th