CMSC206 Review- Fall 2015 – David G. Cooper:

Implicit understanding

Basic/Primitive data types:

- int
- float
- double
- short
- boolean
- char

Java classes Objects:

- every class implicitly extends Object
- define a class with the class keyword
- inheritence
- interfaces

Java API Data type Object Wrappers

- Integer
- Float
- Double
- Short
- Boolean
- Character

Array:

- Like an object.
- Any type can be an array
- Declare by adding [] to the type
- Instantiate with new
- One field: length

Java API: java.util

- interface Collection<E>:
  - boolean add(E e)
  - boolean remove(E e)
  - boolean contains(E e)
  - Iterator<E> iterator()
  - int size()
  - E[] toArray()
  - boolean isEmpty()
- interface List<E> extends Collection<E>:
  - boolean add(int i, E e),
  - E get (int i),
  - E remove(int i)
  - ListIterator<E> listIterator()
- class ArrayList<E> implements List<E>
- class LinkedList<E> implements List<E>
  - boolean addFirst(E e)
  - E removeLast();
  - boolean addLast(E e)
  - E removeFirst()
- class Stack<E>
  - E peek()
  - E pop()
  - E push(E e)
  - boolean empty()
- interface Queue<E>
  - boolean add(E e)
  - boolean offer(E e)
  - E remove()
  - E poll()
  - E element()
  - E peek()

New material

Data Structures:

BinaryTree

BinarySearchTree

Heap/java.util.PriorityQueue<E>

- Array representation
- Indexing: lc = p*2 + 1, rc = p*2+2, p = (c – 1)/2
- Heapify

java.util:

interface Set<E> extends Collection<E>

class HashSet<E> implements Set

class HashMap<K,V> implements Map

- V put(K k,V v)
- V get(K k)

Diagrams

- draw a data structure with it's contents after adding a list of elements in sequential order
- draw a data structure with it's contents after a code snippet has executed
- draw a BST from preorder traversal or post order traversal

Algorithms/Applications

- Searching
  - Items in an unsorted array
  - Items in a sorted array
  - Items in a BST
  - Using java.utils.Arrays
  - Using other java classes

Algorithms/Applications (cont.)

- Expression evaluation
- Simulation (such as printer queue)
- Sorting
  - Insertion
  - Selection
  - Merge
  - Heap
  - Quick
- Recursion
  - Ex. given a base case(s) and a recursive case, write the recursive function. For example: write a recursive function fib, that returns an int
    - base cases: fib(1) = 1, fib(2) = 1
    - recursive case: fib(n) = fib(n-2) + fib(n-1)
- Trees
  - Expression trees
  - Traversals
  - BST
  - Heaps/Priority Queues
- Hashtables
  - Open adressing/linear probing
  - chaining
- Time complexity
  - On loops
  - On recursive algorithms
  - On tree methods
  - Of data structure methods