Java God Sheet

data structures

- LinkedList, ArrayList
    - add(Element e), add(int idx, Element e), get(int idx)
    - remove(int index)
    - remove(Object o)
- Stack
    - push(E item)
    - peek()
    - pop()
- PriorityQueue
    - peek()
    - poll()
    - default is min-heap
    - PriorityQueue(int initialCapacity, Comparator<? super E> comparator)
    - PriorityQueue(Collection<? extends E> c)
- HashSet, TreeSet
    - add, remove
- HashMap
    - put(K key, V value)
    - get(Object key)
    - keySet()
    - if you try to get something that's not there, will return null

- default init capacities all 10-20
- clone() has to be cast from Object

useful

iterator

- it.next() - returns value
- it.hasNext() - returns boolean
- it.remove() - removes last returned value

strings

- String.split(" |\\.|\\?") //split on space, ., and ?
- StringBuffer
    - much faster at concatenating strings
    - thread safe, but slower
    - StringBuilder s = new StringBuilder(CharSequence seq)();
    - s.append("cs3v");
    - s.charAt(int x), s.deleteCharAt(int x), substring
    - Since String is immutable it can safely be shared between many threads
- formatting

```
String s = String.format("%d", 3);
"%05d"  //pad to fill 5 spaces
"%8.3f" //max number of digits
"%-d"    //left justify
"%,d"    //print commas ex. 1,000,000
| int | double | string |
| d   | f      | s      |
```

```
new StringBuilder(s).reverse().toString()
int count = StringUtils.countMatches(s, something);
```

- integer
    - String toString(int i, int base)
    - int parseInt(String s, int base)
- array

```
char[] data = {'a', 'b', 'c'};
String str = new String(data);
```

sorting

- Arrays.sort(Array a)
- Collections.sort(Collection c), Collections.sort(Collection l, Comparator c)
    - use mergeSort (with insertion sort if very small)
- Collections.reverseOrder() returns comparator opposite of default

```
class ComparatorTest implements Comparator<String>
    public int compare(String one, String two) //if negative, one comes first
class Test implements Comparable<Object>
    public int compareTo(Object two)
```

exceptions

- ArrayIndexOutOfBoundsException
- `throw new Exception("Chandan type")`

higher level

types

- primitives - `byte, short, char, int, long, float, double`
- java only has primitive and reference types
    - when you assign primitives to each other, it's fine
    - when you pass in a primitive, its value is copied
    - when you pass in an object, its reference is copied
        - you can modify the object through the reference, but can't change the object's address

garbage collection

- once an object no longer referenced, gc removes it and reclaims memory
- jvm intermittently runs a mark-and-sweep algorithm
    - runs when short-term stuff gets full
    - older stuff moves to different part
    - eventually older stuff is cleared

object-oriented

```
| declare | instantiate | initialize |
| Robot k | new         | Robot()    |
```

- class method = static
    - called with Foo.DoIt()
    - initialized before constructor
    - class shares one copy, can't refer to non-static
- instance method - invoked on specific instance of the class
    - called with f.DoIt()
- protected member is accessible within its class and subclasses