3/27/2019 Notes

* An abstract data type (ADT) is a model for data types
* A data type is defined by its user of the data, values, operations on data of this type, behavior, etc.
* A data structure is a way of organizing data
* Data structures can implement one or more articular abstract data types
* Interfaces control the structure of subclasses
  + Final keyword in an interface is not required because there are no variables in instances
* FullTimeEmployee class is on Chapter 1 Page 33
* When you create objects, the variable isn’t an Object, it is a reference to an Object. Referenced by addresses
* Using the default constructor for String, the string will be “”, not NULL like most objects
* String will return -1 if the subindex does not exist
* String s = new String();
* String t = new String(“Aloha”);
* System.out.println(t.indexOf(“ha”));
  + 3
* System.out.println(t.indexOf(“a”));
  + 4
* System.out.println(s.indexOf(“ha”));
  + -1
* String w = null;
* This does not make an empty string. This will just save a null object
* System.out.println(w.indexOf(“ha”));
  + Exception in thread “main” NullPointerException
* equals() tests comparison of objects
* The == operator tests for equality of references
  + int and boolean only really work
* String s = new String();
* String t = new String(“Aloha”);
* String w = null;
* String z= new String(“Aloha”);
* s.equals(“”);
  + true
* s == “”
  + false
* t.equals(“Aloha”);
  + true
* t == “Aloha”
  + false
* t.equals(null);
  + false
* t.equals(z);
  + true
* t == z
  + false
* w.equals(null)
  + NullPointerException
* w == null
  + true
* Scanners
  + Scan keyboard input, file input, or a given string value
* Arrays
  + Arrays start from zero lol
* Arguments and Parameters
  + Arguments are the data
    - All arguments are sent by value, even if they are references to Objects
  + Parameters are what is in the header
* Review chapter 0 of the book for dumb basic CS110-111 stuff