ETEC2101	Test I "Study Guide"	Test is 10/10/2017

Major Topics (everything we covered in class is fair game, though)

- **Section1: Java crash-course**
  - Connection and role of the JVM, JDK, and JRE and the java build / run process.
  - Printing in Java (System.out.print[ln])
  - Structure of a Java program (including main)
  - Variable declaration (and the major types)
  - Role of new and connection to Objects
  - Object-types vs. Primitive-types (also size of major primitives)
  - Casting (and the reason you need it)
  - Scanners
  - For loops
  - Arrrays in java
  - Static methods
  - Generating random numbers (this was on the lab pdf)
  - BubbleSort (this was on the lab pdf)
- **Section2: OOP and Slick2D**
  - Relationship between classes and object
  - Instance vs. static members
  - Basic role of access-modifiers (protected, private, etc.)
  - Packages
  - Constructors
  - This
  - Interfaces
    - Creating them
    - Implementing them in another class
    - When to use them
    - Limitations
  - Extends inheritance
    - The mechanics / syntax
    - Limitations
    - Abstract methods / classes and extends
  - Polymorphism
  - Slick2d's use of extends and implements
  - Observer design pattern and its implementation in Java
  - Copying (sections) of one array to a different (or the same) array using System.arraycopy (this was on the lab pdf)
- **Section3: ArrayLists**
  - The use of the major methods we implemented
  - Using Iterators
  - Inheritance "tree" of types (slide 8)
  - Throwing and catching exceptions
  - Enumerations
  - Generics (including a "minimum" base-class)
  - Comparing Comparables
  - BinarySearch algorithm
- **Section4: LinkedLists**
  - Advantages (and disadvantages), compared to ArrayLists.
  - Nested classes
  - Connection between code / logical view / memory-view of an existing linked list.
  - Implementation of all major methods.

Exam Format:

- ~50% of the points will be on pencil-and-paper
  - No books, no notes, no computer
  - There will likely be (small) programming snippets using Java
- ~50% of the points will be based on a small program that you'll complete on the computer
  - You have access to google, past labs, etc. (just not your neighbor☺)
  - You'll be able to see the problem as soon as you get the test packet, but you won't be able to use your computer until you turn in the first part.
  - You'll submit the second part on blackboard.
- Make sure to budget your time – I'll have to cut you off at 2:05pm
- 100 points, but it's not on the same "scale" as lab points.
  - (from the syllabus) Final% = 0.5 * Lab% + 0.15 * Test1% + 0.15 * Test2% + 0.2 * Final%
  - On Blackboard, your final grade right now is
    - Final% = Lab%
  - After we have Test1, it will be:
    - Final% = (0.5 * Lab% + 0.15 * Test1%) / 0.65 ≈ 0.77 * Lab% + 0.23 % Test1%

Sample Questions

1. [Pencil/Paper part]
   a. (**10 points**) Describe / show a concise example of polymorphism in action. You can do this with Java code (less ambiguous) or very-carefully-worded English.
   b. (**10 points**) Show an example of using a Scanner to ask the user to enter a line of text (followed by enter), followed by two integers. Print out the line of text and the sum of the two numbers. Show a complete java program, including the main program.
   c. (**15 points**) Discuss (preferably with code snippets) the changes necessary to our linked list so that there is only one add method which adds items to the linked list in ascending order.
2. [On-computer]
   a. (**15 points**) Ask the user to enter a string, determine if this string is a palindrome or not (i.e. the same backwards and forwards) [Note: I probably couldn't ask this question since I quickly found an answer on google – but this about the difficulty for a 15-point problem]
   b. [NOTE: I *could* ask a bunch of small mini-problems like 2a, or one giganto problem worth 50 points]