ENVIRONMENT:

OS: Linux
IDE: JGrasp, GEdit, Eclipse
JDK: 8.0 (1.8.x)
Duration: 4 hours
No: Internet, notes, or text
Java API provided
The exam uses Test Driven Design, where you are writing code based on the provided tests.

SECTIONS:

General Program Design: 30%

- Penalty points can be assessed in this category for grossly inefficient code
- File input and output
- Exception handling
- Order-related manipulations, e.g. ***addOrdered***, ***sort***, ***search***

Data Abstraction and Class Design: 30%

- Creating a class
- Extending a class
- Overriding a method, e.g. ***toString***
- Accessing super class methods and/or fields
- Implementing an interface, e.g. ***Comparable***

Linked List Manipulation: 20%

- Creating a non-circular singly linked list, with or without a header/sentinel node
- Adding and removing nodes
- Retrieving data based on an index or conditions
- Order-related manipulations
- Generate a sub list based on conditions

Recursion: 20%

- Simple recursive methods to accomplish a task
- Printing a linked list in reverse
- Computation of factorials
- Any recursive algorithm covered in CSCD 210, 211, or 300
- Example: Additive Squares

Expanded Specifics

An expanded list of topics follows. You should make sure you are proficient in each area to ensure success.

- **File input and output**
  - Input via a Scanner object

- **Input files will be well formed**
- **know how to stop when end of file is reached**
- **remember to close file when finished**
  - Output via PrintWriter
- **know how to format output (decimal points, columns, etc.)**
- **remember to close output file when finished**
- **Inheritance**
  - Understand base / derived relationship
  - know how to override methods to enable polymorphic behavior
  - understand abstract classes and interfaces
  - know how to derive a class from an existing class
  - know how to call base/parent/super class methods (requires use of keyword super)
- **Comparable interface**
  - know how to implement for a given class, including multiple-key comparisons
  - know how to utilize when searching and sorting (i.e. call the compareTo method)
- **Basic exception handling**
  - know how to designate a section of code as possibly throwing an execption (via a try block)
  - know how to handle and exception (via a catch block)
  - know how to propagate an exception from a method (by using throws and the end of the method signature)
  - know how to generically handle an exception (use Exception to do this)

Frequent Problem Areas

- Not following directions
- Not reading the specifications in their entirety
- Ignoring specifications related to header/sentinel nodes
- Not placing files in specified location
- Not naming files as specified
- Modifying files that were to be left alone
- Unfamiliarity with keywords in Java required for basic things like inheritance (extends) and interface use (implements)
- Spending too much time on one item
- Inability to implement simple recursion

Basic Strategies

- Stub all required methods, then fill them in as you progress through the exam
- Carefully read the directions, and look at the sample input and output before you begin writing code
- Temporarily comment out items that don't compile so you can test parts of your solution

Practice Exams

Winter 2011: Exam | Solution
Winter 2012: Exam
Summer 2013: Exam
Summer 2017: Exam | Solution

Keep in mind that these are just sample exams. Do not expect the exam you take to be the same. The amount of code you are asked to write on the practice exams is a good reference for the actual exam.