Final Review Topics

- Chapters 1-9 of the textbook, including (but not limited to):
  - Java language syntax and semantics.
  - Definition & usage: classes, objects, methods/behaviors (accessors, mutators), instance/field variables, messages, new object instantiation construction/destruction, object state
  - Inheritance: base/super classes, derived/sub-classes, abstract classes/methods, super constructor-method invocations, extending, access levels: (private, protected, public), polymorphism, over-ridden methods, type-casts, static/dynamic type, method dispatch/lookup
  - Primitive Data: arithmetic expressions, boolean expressions, String expressions, precedence
  - References / Aliases, null, Equality, method parameters, return values, void
  - Variable Scope: parameters, local, instance/field, this
Final Review Topics (continued)

- Selection statements: if, if else & switch
- Repetition: for, while, nested
- Recursion: finite, iterative equivalence, infinite
- Pre-defined types: Strings, StringTokenizer
- Containers/collections: ArrayList, Vector, arrays, Map/HashMap, HashSets, (Iterators, Enumerations, indexing)
- I/O: BufferedReaders, PrintWriters, System.out, IOHelper
- Static: methods, class constants (final)
- Interfaces: implements
- Testing: writing your own unit tests
- No Karel material
Test 1 Objectives

Learning Objectives (Karel knowledge NOT included):

- Trace, (desk check, hand trace) the execution of small Java programs
- Understand Java syntax and semantics
- Know the basic Java reserved words
- Define Java identifiers, (references, variables)
- Know how to invoke class methods and pass parameters to them
- Create reference variables and utilize them to pass messages, set aliases and change references
- Apply Java assignment statements
- Know the different Java comment syntaces
- Recognize equivalent statement sequences
- Instantiate objects using the new operator, initialize with constructors and store their references
- Use cascading and composition to send messages to objects
- Invoke standard Java language methods
- Know how to import Java packages
Test 1 Objectives (continued)

- Learning Objectives (*Karel knowledge NOT included*):
  - Know when class constructors are invoked
  - Be able to instantiate new Java objects
  - Design, code and invoke methods in Java classes
  - Declare method return types and utilize return statements
  - Understand and apply class instance/field variables
  - Recognize which access specifiers, (public & private), to apply where/when in a class
  - Use return variables in Java statements and expressions
  - Know when a method should return void or null
  - Distinguish between (and employ) the three types of variables in Java: (parameters, local variables & instance/field variables)
  - Define and evaluate Java Boolean expressions
  - Apply & trace the Java selection statements: if, if else
  - Compose nested Java selection statements
  - Identify & implement simple recursive methods: finite and infinite
Test 2 Objectives

Learning Objectives

- Utilize the pre-defined Java language System.out output stream
- Instantiate BufferedReader, and PrintWriter objects using the CS 1705 package IOHelper class
- Utilize & understand the basic Java input/output stream member methods, (i.e. know the model of Java streams)
- Trace the execution and output of Java file stream code
- Identify and code overloaded methods in Java classes
- Know how long, (in terms of code execution), variables and objects exists in a Java program
- Define, form and evaluate expressions with Java primitive variables
- Utilize the Java arithmetic operators in expressions
- Apply the Java increment and decrement operators
- Know the difference between reference comparisons and object comparisons
- Be able to read and understand Java API documentation
Test 2 Objectives (continued)

- Learning Objectives
  - Decide when the null reference should be returned by Java methods.
  - Know the general terms that apply to programming iteration statements: identify the loop control variable and its initialization, updating and testing loop expressions.
  - Compose, trace and understand Java code employing the while/for loop statements.
  - Determine when a loop would result in infinite execution.
  - Employ Boolean variables to compose loop termination conditions.
  - Create function definitions and function method declarations.
  - Distinguish and understand the matching of the formal and actual function method parameters.
  - Determine the scope of identifiers by their location.
  - Store a list of data in a Vector, ArrayList & array.
  - Understand the difference between containers and simple data types.
  - Use and employ standard Vector, ArrayList & array terminology.
Test 2 Objectives (continued)

- Learning Objectives
  - Define a Java Vector, ArrayList & array object for a given data set
  - Access and initialize the individual elements of a Vector, ArrayList & array
  - Distinguish between the usage and size of a Vector, ArrayList & array
  - Pass Vector, ArrayList & array and vectors elements as parameters to functions
  - Implement operations on Vector, ArrayList & array
  - Trace Java code involving Vector, ArrayList & array
  - Recognize out-of-bounds Vector, ArrayList & array index accessing
  - Know how to define an inheritance relationship between two classes
  - Understand the definitions and usage of inheritance terminology
  - Be able to invoke the base constructor from a derived class constructor
  - Know and understand the three ways to indicate class members access: private, public, protected
Final Exam Objectives

- Learning Objectives
  - Know how to utilize iterator objects to cycle through a container’s objects
  - Compose, trace and understand Java code employing the following StringTokenizer class functions:
    - `hasMoreElements()`, `nextElement()`, `nextToken()`
  - Be able to answer the key questions of responsibility-driven design for simple programs (a class should be responsible for managing its own data):
    - Which class “owns” the data of interest? What methods should it provide to manipulate this data?
  - Be able to identify poorly designed code that exhibits tight coupling, loose cohesion, or duplicated code
Learning Objectives

- Code, trace, modify and perform the Selection sort algorithm.

- Create, trace, and modify Java code for performing a sequential, linear, search on un-sorted and sorted lists of data.

- Create, trace, and modify Java code for performing a binary search on lists of data.