Test 2

- Chapters 1-4 of the textbook

- Topics:
  - Primitive Data: arithmetic expressions, boolean expressions, String expressions, precedence
  - Containers: ArrayList, Vector, arrays
  - References / Aliases, Equality
  - Variable Scope: parameters, local, instance/field, this
  - New object instantiation
  - I/O: BufferedReader, PrintWriter, System.out, IOHelper
  - Repetition: for, while, nested
Test 2 Objectives

Learning Objectives

- 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 syntaxes.
- 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.
Test 2 Objectives (continued)

- 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.
  - Know when class constructors are invoked.
  - Be able to instantiate new Java objects.
  - Design, code and invoke methods in Java classes.
  - Identify and code overloaded 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.
  - Distinguish between (and employ) the three types of variables in Java: (parameters, local variables & instance/field variables).
  - Know how long, (in terms of code execution), variables and objects exists in a Java program.
Test 2 Objectives (continued)

- Learning Objectives
  - Define, form and evaluate expressions with Java primitive variables & Strings.
  - Utilize the Java arithmetic operators in expressions.
  - Apply the Java increment and decrement operators.
  - Define and evaluate Java Boolean expressions.
  - Apply & trace the Java selection statements: if, if else.
  - Compose nested Java selection statements.
  - Know the difference between reference comparisons and object comparisons.
  - Recognize when the null reference may be returned by Java methods.
  - Return a null reference when it is appropriate.
  - Know the general terms that apply to programming iteration statements: identify the loop control variable and it's 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.
Learning Objectives

- 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.
- 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 an 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.