The test has 4 questions: Executing sequence of assignments, writing equals functions, writing constructors, and knowing definitions/concepts. To pass the test, you must demonstrate that you understand points 1..3. Below, we give information on each of 1..3. We will not give these tests ad infinitum. We will give the test the week before Fall Break, and then give up to two more retests at scheduled.

1. **Sequence of assignments.** The purpose of the question is to ensure that you know:
   1. How to execute an assignment statement (evaluate the expression and store the value in the variable) and you can do it as the computer does. *Do not draw a new variable; cross off the old value and write in the new one in the single box denoted by the variable name.*
   2. How to evaluate a new expression, including knowing what the value of the new expression is.
   3. That a new object is drawn *only* when a new-expression is evaluated, and every time a new-expression is evaluated, a new object is drawn.
   4. How to draw an object, putting in each partition the non-static variables and methods declared in the class for that partition.

**What to do to be able to answer this question correctly:**
   1. Memorize how to evaluate a new expression — don’t just read it, write it down.
   2. Look at our solution to one such question and (1) see how we draw variables, (2) see how we draw objects, (3) see how we store a value in a variable.

2. **Writing equals functions.** The purpose of this question is to ensure that you:
   1. Can read and interpret a specification correctly. *Always read the specification and implement it!*
   2. Understand and can use the `instanceof` operator.
   3. Know about casting from one class to another and understand why it generally has to be done in an equals function.
   4. Can write a return statement that returns true if one or more variables have the right values
   5. Understand that `private` variables of a superclass cannot be referenced directly in a subclass.

We give you at least one equals function that is properly written. You will be able to write similar ones without difficulty if you can handle point 1-5 above.

**What to do to be able to answer this question correctly.**
   1. Read about and memorize the `instanceof` operator.
   2. Learn about casting from one class to another, what it means and how to do it.
   3. Study the properly written equals function that we give you.
   4. Practice writing equals functions – compare your answers to the sample one. Write and test them in DrJava.

3. **Writing constructors.** The purpose of this question is to ensure that you know:
   1. That first statement of a constructor *must* be a call on another constructor. If you don’t put one in, Java inserts “super();”.
   2. This Principle: When initializing the fields of an object, initialize the superclass fields first.
   3. That, generally, the best way follow point 2 is to use a call on a constructor in the superclass, of the form “super(…);”. To get that call right, you have to look at the specification of the superclass constructor.
   4. This Principle: If at all possible, rely on previously written methods rather than duplicate code, saving you time in programming and debugging.
   5. That sometimes, in a constructor, you can use point 4 by calling another constructor in the same class, using a call “this(…);”. To get the call right, you have to look at the specification of the other constructor.
   6. That when writing a constructor, *draw an object of the class, making sure to put in all fields! This will help grasp what has to be done. We cannot stress this point enough, although none of you seem to want to do it.*

**What to do to be able to answer this question correctly.**
   1. Study carefully point 1-6 given above and look at our sample solution to one problem.
   2. Practice more problems like the sample one.