## CS1110 About Prelim III: Tuesday, 14 April, 7:30–9:00PM, Ives Hall 305
## Review session: Sunday, 12 April, 1:00–3:00, Phillips 101.

For this prelim, you need to know everything that you needed to know for the first two prelims (see review below). The handout for Prelim II (which you can get from the course webpage) contains a list of topics as well as a collection of concepts and definitions. These deal with classes, subclasses, the class hierarchy, inheritance, overriding, casting, operator instanceof, abstract classes, kinds of variables, methods and calls on them, executing method calls, function equals, wrapper classes, etc.

In any course such as this one, the material is cumulative, and you need the old material to understand and use the new material. Therefore, all material in the course is fair game for the prelim.

Besides your normal studying, spend time, frequently, practicing writing and testing Java code; that is the only way to gain real fluency. Many of you restrict your use of DrJava to what is required on the assignments and labs. That is a mistake; you are not required to limit yourself to labs and assignments! For example, if your answers to a sample prelim don't match our solutions, try typing in your (or our) solutions to DrJava and seeing what happens.

Below, we give some more details on what can be on prelim 3:

**For loops**. We may give you a problem that requires you to write a loop (with initialization) that processes a range of integers. You should be able to write a postcondition, then write the loop header "**for** (**int** k ...)", then write a loop invariant, and finally develop the various parts of the loops and initialization.

**While loops**. We may give you a precondition, postcondition, and a loop invariant, and you will have to develop the loop, with initialization, from it. The grade will depend on how well you deal with the four loopy questions and the given invariant. The only variables that should be used (outside of variables declared in the repetend) are variables that are mentioned in the loop invariant.

**Arrays**. Everything on Sects 8.1 and 8.2 of the class text —these pages discuss the technical details for using arrays in Java and for reasoning about arrays, including the range notation h..k (where h..h-1 is allowed, and indicates the empty range). Two dimensional arrays: You should know how to declare and use rectangular arrays *but not ragged arrays*; see Sects. 9.1, 9.2.

**Algorithms**. You should know the following algorithms. For example, if we ask "what is algorithm partition?"; you should be able to write the precondition and postcondition (we won't give them to you), write down the invariant, and then develop the loop with initialization. These algorithms are discussed in the text, on the ProgramLive CD, and in lectures notes:

Algorithm Binary search (Sec. 8.5.3)
Algorithm Dutch national flag (done in lecture)
Algorithm Partition (PLive activity 8-5.5
class notes may be better)
Selection sort (Sec. 8.5.4)
Insertion sort

For selection sort and insertion sort, the repetend should be written in English, stating *what* the repetend does and not *how* it does it. Since you know that at least one of these algorithms will be on the test, practice developing all of them. Answering this question should take 5 minutes, because you *know* how to develop them.

**Classes String and Vector**. We use these all the time. You are expected to know the basic methods of these classes: For class String: charAt(i), substring(i), substring(i,j), length(). For Vector: add(ob), set(i, ob), get(i), size(). If a question calls for other methods of the classes, we will define them for you. You should know that **new** Vector<C>(...) creates a Vector whose elements are of class <C>.

**Exception handling**. Exception handling will *not* be on prelim 3.

**GUIS**. We will not ask you questions about GUIs. They may be on previous prelim 3's but will not be on this prelim 3.

**Reading and writing files.** You will not be asked to memorize how to read and write files. We don't remember either and look it up whenever we have to.