CS1110 About the Final Wednesday, 10 December 7:00–9:30 pm Uris Auditoriu7

Office hours are canceled for the week after classes (beginning 8 December). However, if you want to see a TA, they will be happy to make an appointment with you. To make an appointment with Gries, call Cindy Pakkala at 255-8240.

Review sessions, beginning 7 Dec., expected to be in Phillips 101, but check the course website the morning of the review session.

Day	Time	Instructor	Topic
Sun	1PM	Danaher	Executing sequences of statements that involve creating new objects
Sun	2PM	Danaher	Writing constructors in classes and subclasses. Casting, apparent and real classes
Sun	3PM	Chan	Arrays
Mon	1PM	Gries	Developing loops from invariants
Mon	2PM	GutFraind	Exception handling
Mon	3PM	Tettemer	Specific classes: e.g. Integer, Character, Vector, String
Tue	1PM	Gries	GUIs, abstract classes
Tue	2PM	Evans	Executing method calls; frame for a call
Tue	3PM	Gutfraind	Recursion

The final is cumulative, *covering all topics in the course* except as described below. So, you have to know everything that was covered in the three prelims. See the handouts on the three prelims (on the course web page).

You do not have to study the following topic: reading a file or the keyboard, applications, applets.

You do have to know about:

1. **Several algorithms**. You know this already, but we repeat it for emphasis. One of the following algorithms can be asked for. We may simply write “show binary search”, or “Show us the partition algorithm”, and you have to give the precondition, postcondition, and loop invariant and then develop the algorithm. We expect that: the loop with initialization is developed from the invariant; a loop that has nothing to do with the invariant gets little credit. Everyone should get full credit on this question because it is simply a matter of (1) memorizing specifications and then (2) practicing developing known algorithms from their specs. For selection sort and insertion sort, write only a single loop, not a nested loop, as explained *ad nauseum* in lectures and in powerpoint slides, and the text.

Linear search, Binary search, Dutch National Flag, Partition algorithm, Selection sort, Insertion sort.

2. **Developing an algorithm: stepwise refinement**. We have used stepwise refinement in class many times, attempting to solve a little bit of a problem at a time. Read Sec. 2.5 on p. 82, and you might also study Sec. 9.2, p. 304, which discusses the development of several problems that deal with arrays.

3. **Arrays**. You have to know about one-dimensional and rectangular arrays. This includes knowing how to access the number of columns in a row and knowing how to create a rectangular array. You have to know how arrays are stored as objects (folders) and to be able to draw an array.

4. **Exception handling**. Be able to write a class that extends Throwable, Exception, or RuntimeException, including the two constructors. Be able to write code to create an instance of such a class and throw it. Be able to write a simple try-statement with a single catch-clause. Understand what happens when an exception is thrown. Do not concern yourself with the *throws* clause; you won’t need it. See the chapter on Exception Handling.

5. **Abstract classes**. Know the purpose of making a class abstract and the syntax for it. Know the purpose of making a method abstract and the syntax for it. See Sec. 4.7 of the text and lesson page 4-5 of the ProgramLive CD.

6. **Placement of components in a GUI**. The default layout managers for a JFrame, a JPanel, and a Box and how that manager arranges components in it. What these basic components are: `JButton`, `JLabel`, `JTextField`, `JTextArea`. Be able to understand programs that place components in a GUI and the code for listening to an event. You do not have to write code.