Office hours are canceled for the week after classes (beginning 5 May). 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, week of 5 May, all in Phillips 101

The order of the topics may change somewhat, but we will put the final schedule on the course website.

Day	Time	Instructor	Topic
Mon	1PM	Wang	Executing sequences of statements that involve creating new objects
Mon	2PM	El Hawary	Writing constructors in classes and subclasses
Mon	3PM	Worthington	Casting, apparent and real classes
Tue	1PM	Gries	Developing loops from invariants
Tue	2PM	Cheng	Interfaces
Tue	3PM	Ramanujan	Specific classes: e.g. Integer, Character, Vector, String
Wed	1PM	Levitan	Exception handling; GUIs
Wed	2PM	Rong	Executing method calls; frame for a call
Wed	3PM	Grayson	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 topics: abstract classes, 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 practicing developing known algorithms from their specs.

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. **Multi-dimensional arrays**. You have to know about 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.

You should also know how multidimensional arrays are handled and how this lets us have ragged arrays.

4. **Exception handling**. We will not expect you to write a complete program that deal with exception handling. However, you should know the basics: What an Exception is, how one creates an object that can be thrown, the try statement, the throw-statement, what happens when an object is thrown —where is it thrown to and how can it be caught, etc. Prelim 3 contained a question on exception handling, and you can expect a question on the final at the same level of understanding. This is explained in the chapter on Exception Handling.

5. **Interfaces**. You should know the basics of interfaces —how they are defined/declared; what their purpose is; how one indicates that a class implements an interface and what that means. You are responsible for Sections 12.1 and 12.2, but not 12.3 and 12.4.

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. You do not have to write code to "listen" to an event, but you should understand how interfaces (see point 5) are used in doing this and you should be able to understand code that does the listening.