

Review sessions, week of 3 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		Executing sequences of statements that involve creating new objects
Mon	2PM		Writing constructors in classes and subclasses
Mon	3PM		Casting, apparent and real classes
Tue	1PM	Gries	Developing loops from invariants
Tue	2PM	Gries	Developing required algorithms
Tue	3PM		Exception handling; GUIs
Wed	1PM		Specific classes: e.g. Integer, Character, Vector, String
Wed	2PM		Executing method calls; frame for a call
Wed	3PM		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 have to know:

1. **Multi-dimensional arrays.** See below.

2. **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 know how to “listen” to an event.

3. **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.

5. **Developing an algorithm: stepwise refinement.** On prelim 3, we had that little problem of writing a function to create Pascal’s triangle in an array. You can expect something similar on the final: a problem that asks you to develop an algorithm/program. The way to get these right is to use “stepwise refinement”, attempting to solve a little bit of the problem at a time. We discussed this in class many times. 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.

6. **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 do not have to know about ragged arrays (in which rows may have different lengths).

7. **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. This is explained in the chapter on Exception Handling.