

A Little About Objects and Classes A possible source of confusion at this point — some things (e.g., description of packages from last time, Scala API) only make sense if you know a little about objects and classes. Key ideas of "object-oriented programming", a major topic in follow-on course CSCI 1321. History of object-oriented programming goes back to (relatively!) early work on simulating physical systems with programs. If writing such a simulation, useful to have a nice way in the program to represent physical objects being simulated (e.g., a car in a simulation of traffic) and groups of similar objects (e.g., cars — all distinct but have common features). "Object-oriented programming" — generalize this idea to more abstract things (e.g., rational numbers, components of a GUI). Key idea is that objects somehow encapsulate both data (variables) and functionality (code).

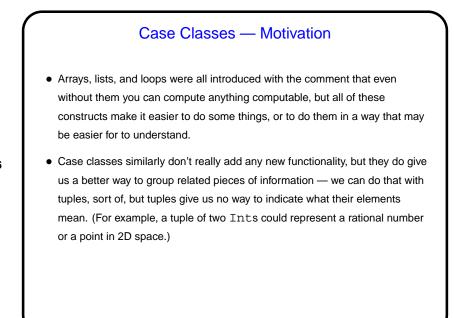
Slide 4



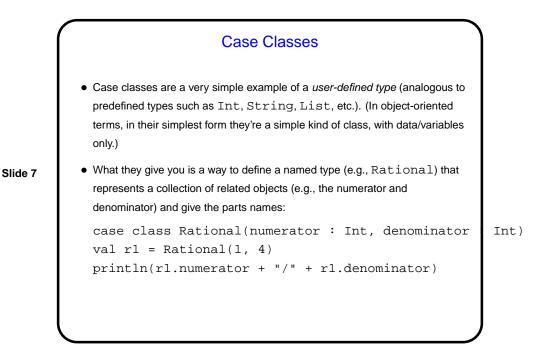
• Scala classes represent different types — simple ones such as Int and String, more complex ones such as io.Source and java.io.PrintWriter. Class definitions (somewhat analogous to function definitions) include variables and "methods" (functions that work on instances of the class).

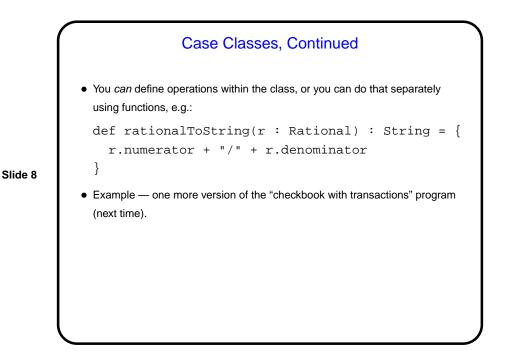
Slide 5

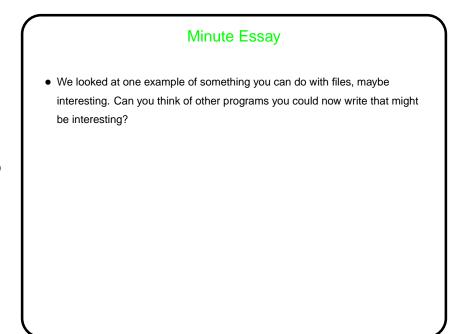
- Scala objects are either instances of a class (e.g., a single Int or String) or singleton objects such as math. Definitions of singleton objects can include variables and methods.
- Keyword new creates an instance of a class.
- Syntax for accessing/invoking data and methods uses dot/period (e.g., math.Pi or math.sqrt(2).
- API shows objects and classes (sometimes have both with the same name e.g., Int).



Slide 6







Slide 9