

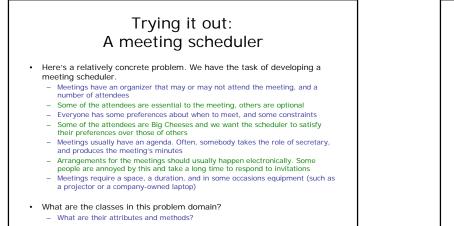
Limitations of the Object-Oriented Paradigm



- Temporal concepts
 - If we represent them with objects, we end up with awkward processes to handle them
 Is a year the accumulation of 365 day objects?
 - Is a year the accumulation of sos day objects?
 If we don't, we may lose other advantages of object-orientation (such as
 - encapsulation)
- · Abstract concepts
 - Can we reduce "preference" to an object or attribute?
- · And sometimes objects just get in the way
 - You need a quick script and Java insists on a full object structure
 - Hence the rise of non-dogmatic object oriented languages (e.g. Python)

ECE450 - Software Engineering II

6



7

ECE450 - Software Engineering II

Some observations... · There was some unnecessary information that should not be modeled · Not all of our classes will become classes in the system - But many of them will • So far we're dealing with the business logic - Later on we should also consider how the system is going to support the methods of these classes User interface objects Databases · Most likely we got some classes wrong and we didn't plan for change - E.g. the meeting is a videoconference and we need to book rooms both in Toronto and in Vancouver · And we're on different timezones - This is where patterns will come in handy PRACTICE THAT EXERCISE IN OTHER DOMAINS - I guarantee you it's going to come up in your exam ECE450 - Software Engineering II 8

