

---

600.325/425

# Declarative Methods

---

Prof. Jason Eisner

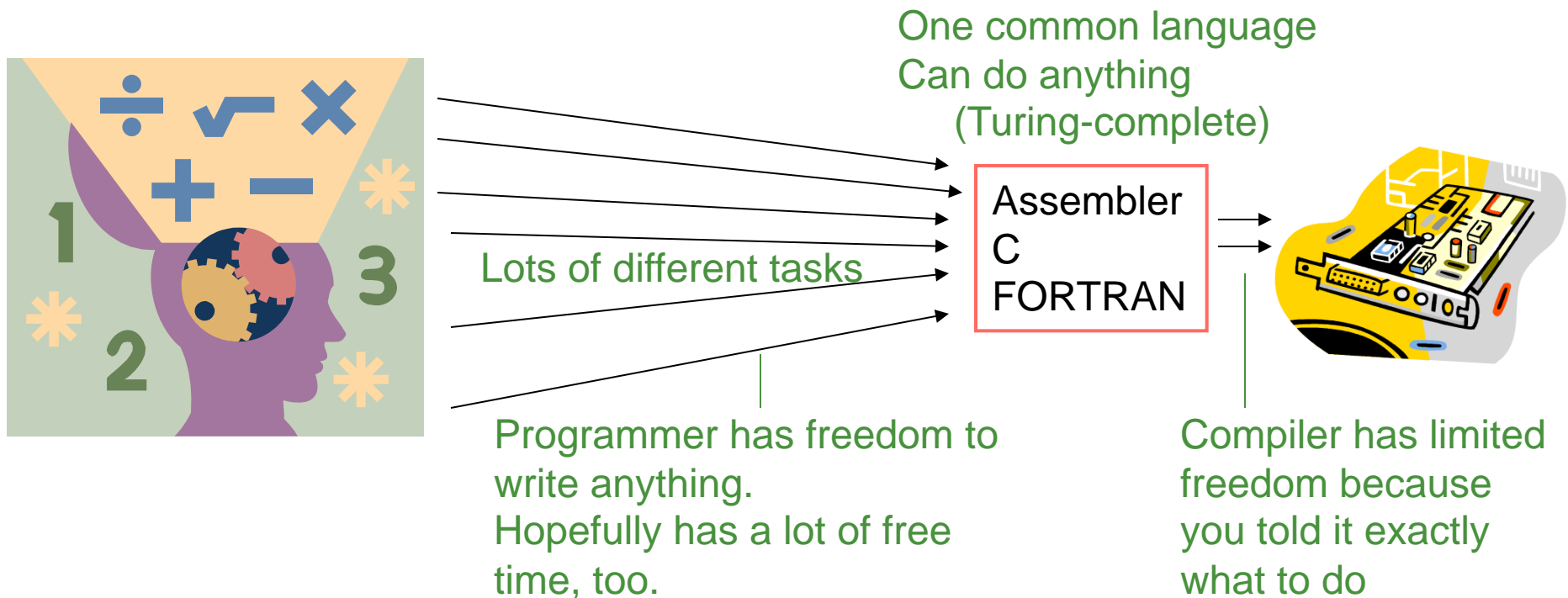
MWF 3-4pm Hodson 311

---

# What is this course about?

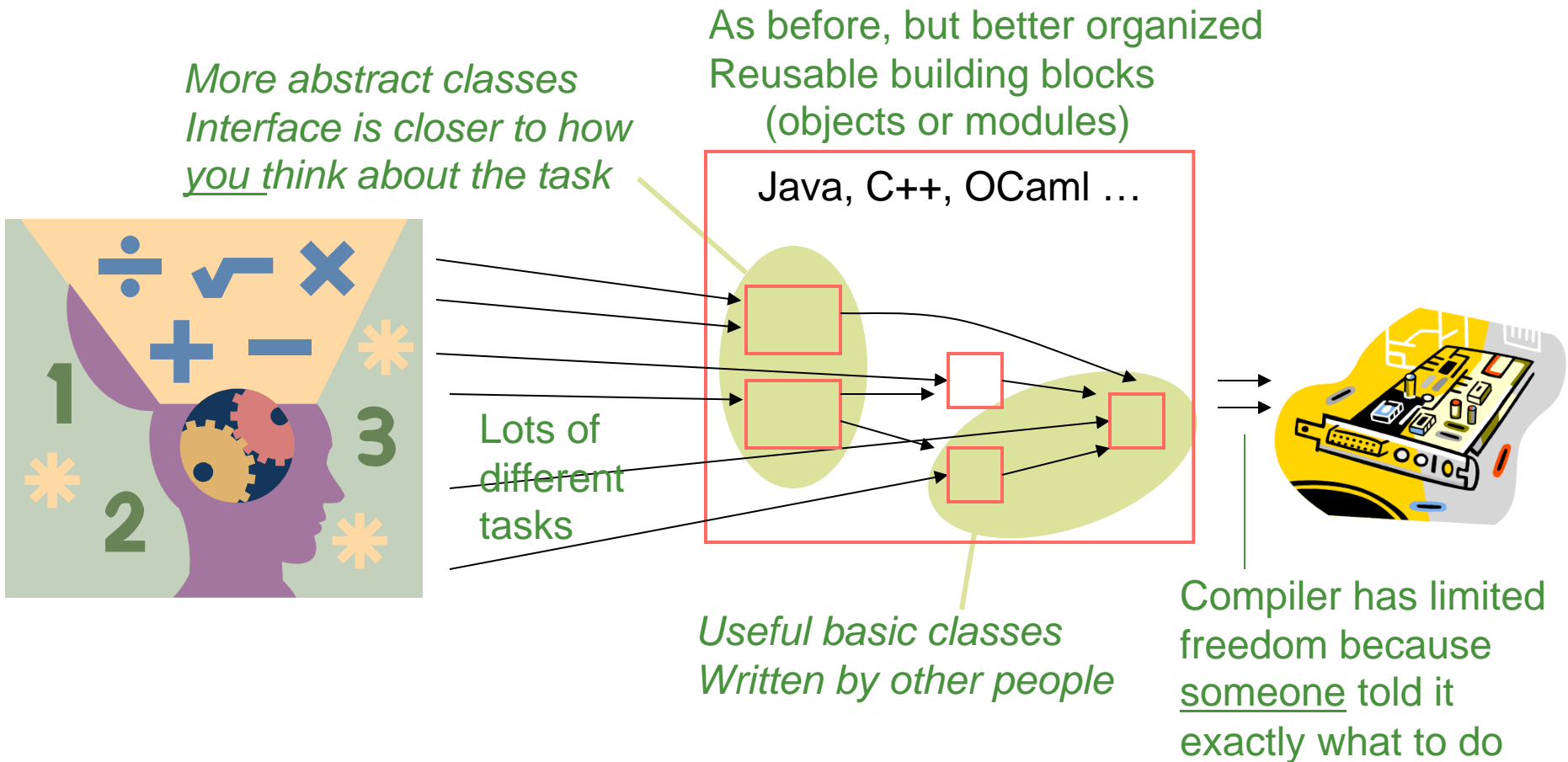
- What do you learn in a programming course?
  - How to use a language (e.g., Java) to solve problems
  - How the computer actually executes that language
    - (Why do you need to know this?)
- Ok, this is a programming course
  - We'll survey several languages
  - But they aren't normal languages!

# Low-level vs. high-level languages



Low-level: Long, detailed programs written by anal-retentive programming gurus

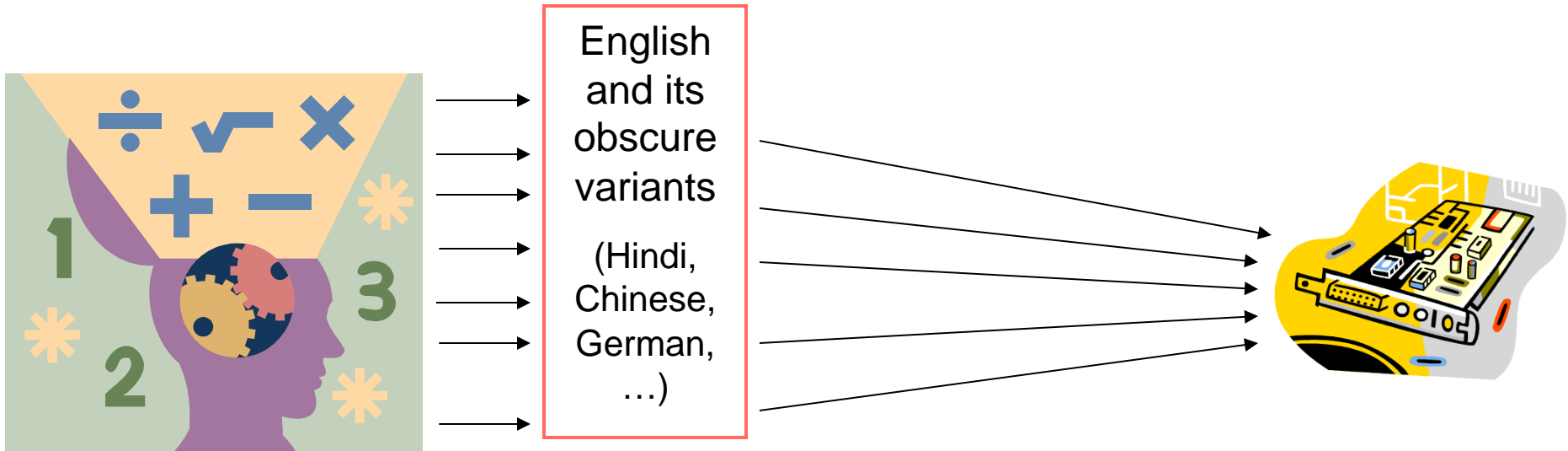
# Low-level vs. high-level languages



Building up high-level objects from low-level ones  
But language and compiler are still low-level

# Low-level vs. high-level languages

A higher-level language  
that can also do anything  
(Turing-complete)

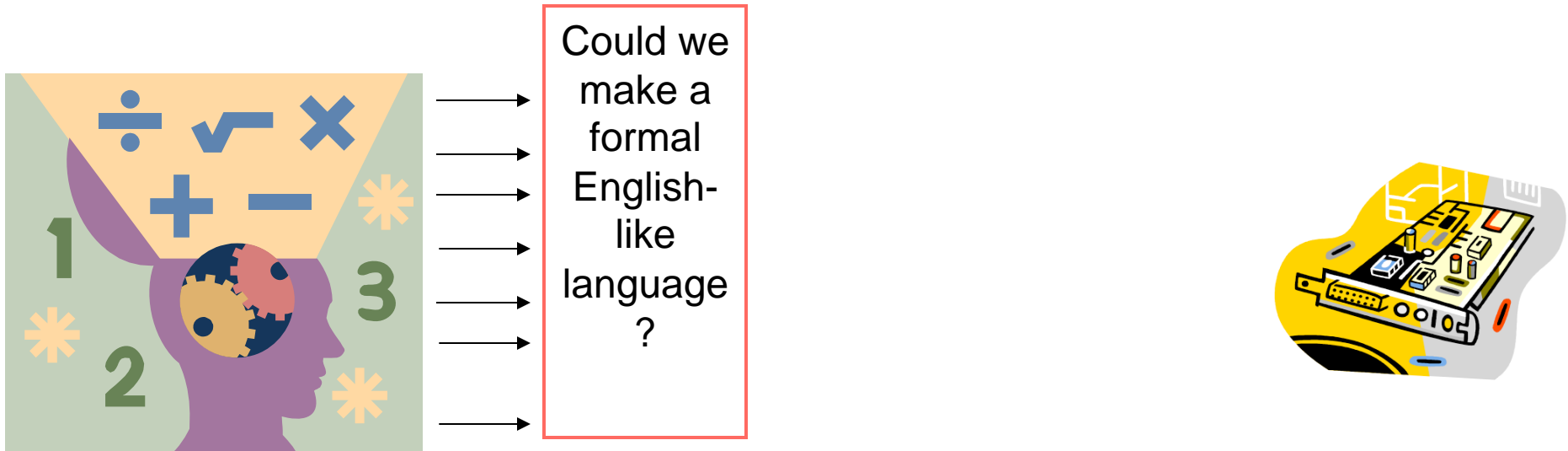


Boy, wouldn't you like to write **this**  
optimizing compiler?  
(take 600.465 NLP)  
(and 600.463 Algorithms)

Really high level! Programming for the masses!

# Low-level vs. high-level languages

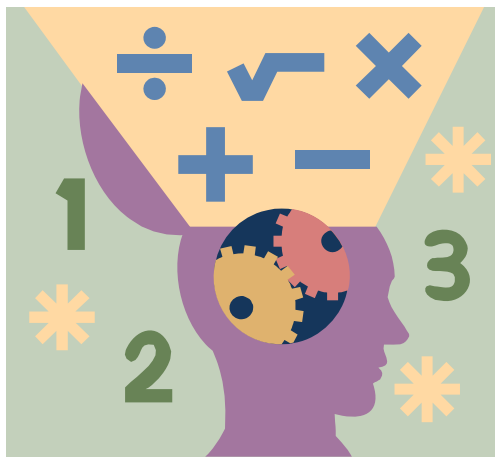
A higher-level language  
that can also do anything  
(Turing-complete)



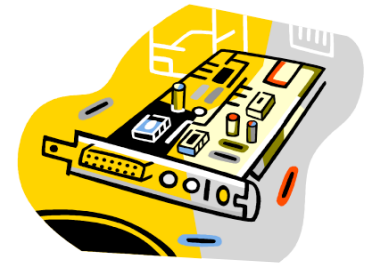
“On each line, replace every third word by x's, then sort the words by length ...”  
It would have to know an awful lot of concepts (line, word, third, sort, length).  
Maybe just make a big library of specialized objects for those concepts?  
Some of those objects would need to have pretty powerful methods:  
“Schedule the classes to minimize time conflicts.”

# Low-level vs. high-level languages

Another language that  
can do anything  
(Turing-complete)



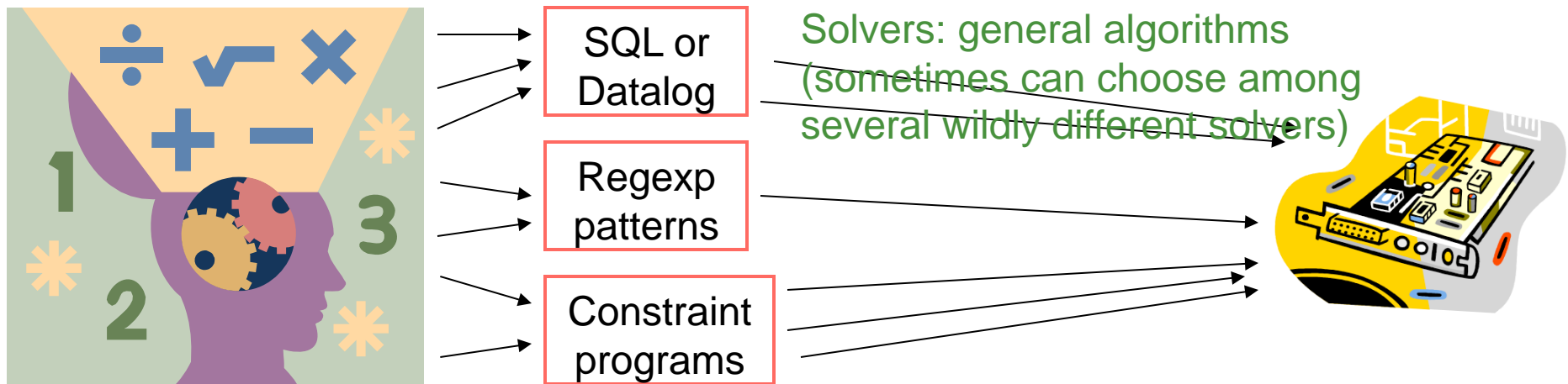
What  
would  
people  
write in  
here  
anyway?



A lot of the same kinds of stuff over and over, actually  
So maybe it *is* good to build some powerful, general, reusable objects to  
handle cases that are either **common** or **hard**  
Then you don't waste your time doing the same kind of thing again & again  
And you don't waste your time figuring out how to do something new & hard

# Low-level vs. high-level languages

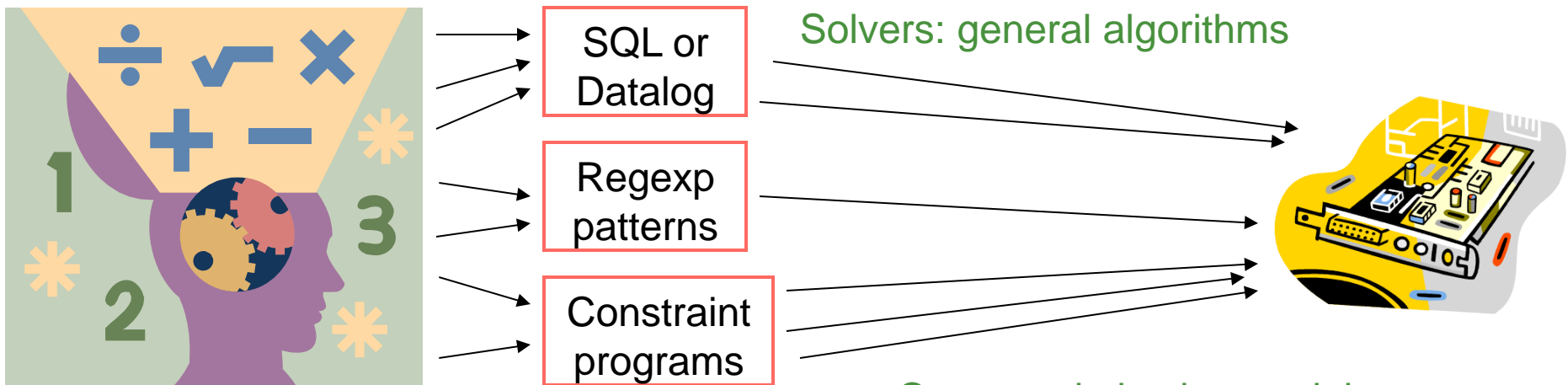
Several specialized  
high-level languages  
“Tools for the job”



A lot of the same kinds of stuff over and over, actually  
So maybe it *is* good to build some powerful, general, reusable objects to  
handle cases that are either **common** or **hard**  
Then you don't waste your time doing the same kind of thing again & again  
And you don't waste your time figuring out how to do something new & hard

# Low-level vs. high-level languages

Several specialized  
high-level languages  
“Tools for the job”



To tell the database about  
your problem, use SQL  
(standard query language).  
More expressive than  
calling a database method!

Query optimization – might  
even compile your query  
into machine code before  
running it