Lies My OO Teacher Told Me

Eric Normand

PurelyFunctional.tv

Registration for Classes

- Go to a department
- Read the list of classes given
- Put your name on the list for each class
- Repeat

Background

- These are lies I have personally experienced in my education.
- Most of my education was in Object Oriented Programming.
- It may seem like I'm attacking Object Oriented Programming.
  - But I'm not. I have a lot of respect for OOP.
- But these are things I've learned through functional programming.
- And they are universally applicable.

For the video and transcript of this presentation, click here:

https://lispcast.com/lies-my-oo-teacher-told-me/

Object Oriented Modeling Process

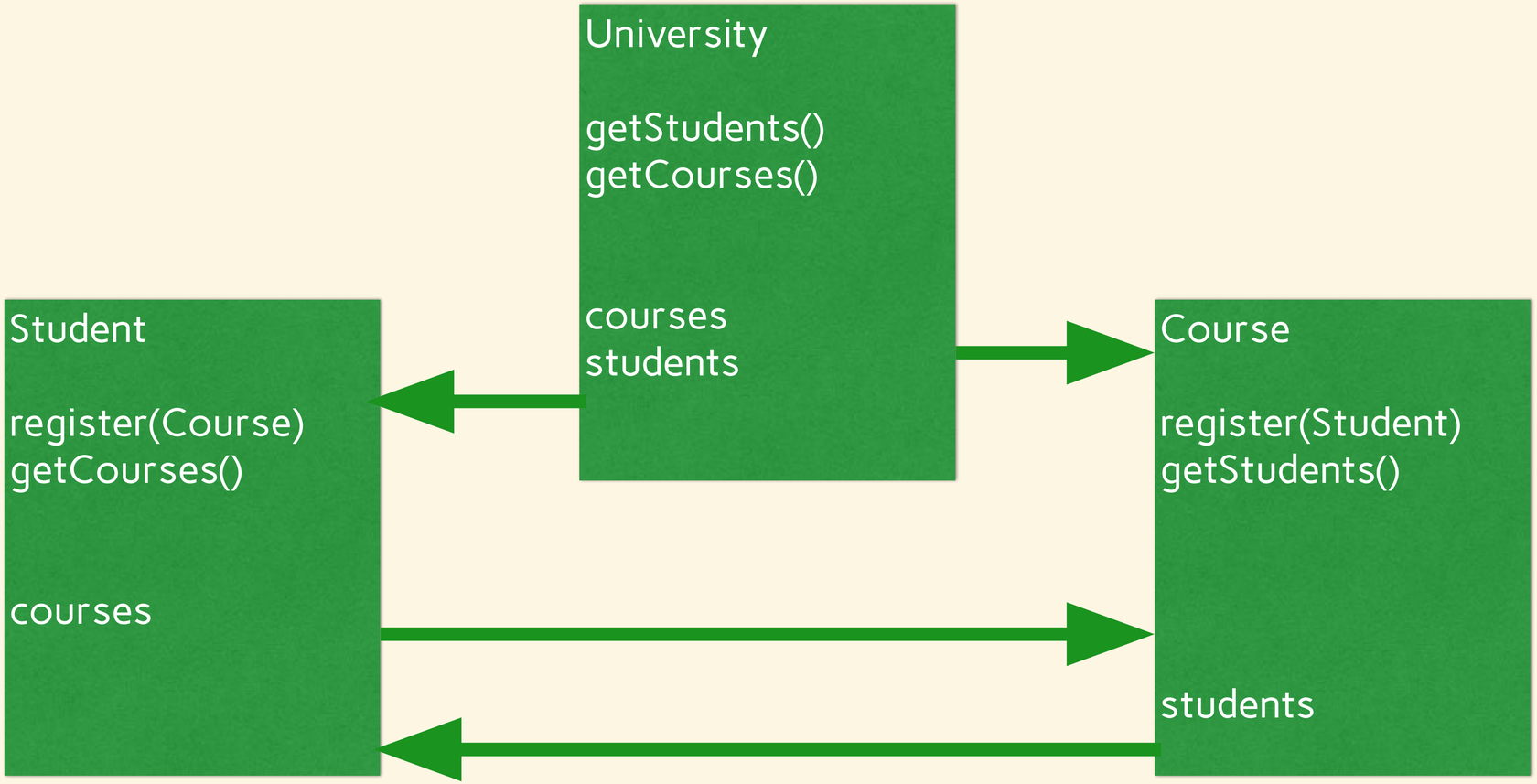
- Take a natural-language description of the problem
- Underline all of the nouns.
  - Make a class for each noun.
  - Look for possessive words like “have”, those become references.
- Underline all of the verbs.
  - Make a method for each verb.

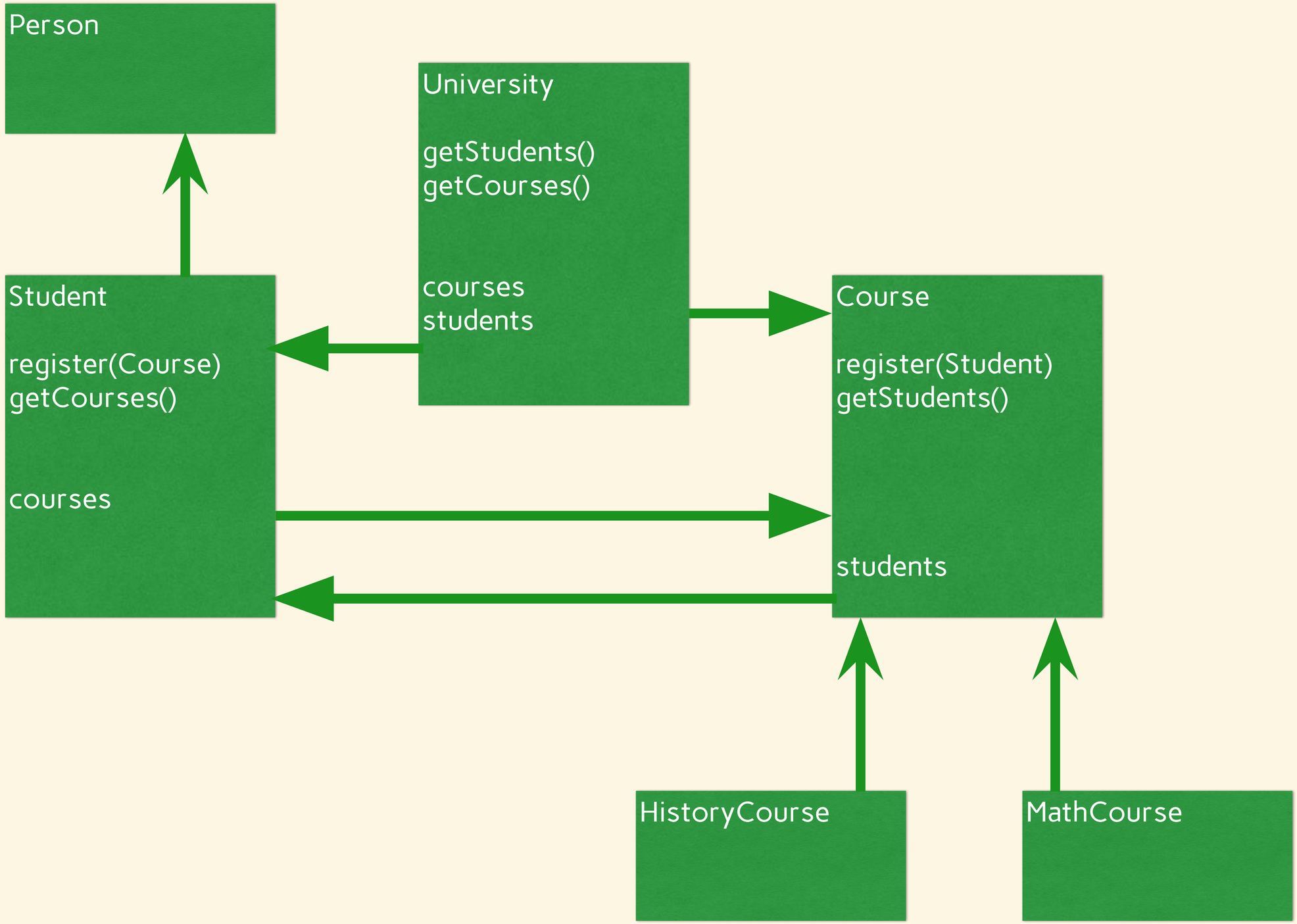
Student Registration

- A university allows students to register for courses. Each student may register for multiple courses. And each course can have multiple students.

Student Registration

- A university allows students to register for courses. Each student may register for multiple courses. And each course can have multiple students.


University
getStudents()
getCourses()
courses
students
Student
register(Course)
getCourses()
courses
Course
register(Student)
getStudents()
students


Person
University
getStudents()
getCourses()
courses
students
Student
register(Course)
getCourses()
courses
Course
register(Student)
getStudents()
students
HistoryCourse
MathCourse

LIES!

Lie #1: Noun and Verb?

- Student
  - Study
  - Studious
- Register
  - Registration
  - Registry

Takeaways

- We have to look deeper than part of speech.
- We have many more options for modeling.

Command-Query Separation

- Queries:
  - Methods that are like questions you ask the object.
  - No modifications.
- Commands:
  - Methods that modify the object or call a command on another object.

Questions - Queries

- What is your name?
- What is your Student ID?
- What classes are you registered for?

Actions - Commands

- Change your name to “xxx”.
- Change your Student ID to “xxx”.
- Register for X course.

LIES!

Lie #2: Those are interesting questions

- These are questions, but do they modify?
  - What would it take to register you for this course?
  - I bet your schedule would be better if you added this course? Is it?
  - What was your schedule last year?
  - I know you haven't decided, but could you give me an example schedule?

Takeaways

- Easy questions like “What is your name?” are uninteresting.
- Interesting questions are more like commands in an alternate reality.
  - Hypotheticals
  - Counterfactuals

Model relationships with references

- Student.courses - list of pointers to Courses
- Course.students - list of pointers to Students
- University.courses - list of pointers to Courses
- University.students - list of pointers to Students

LIES!

Lie #3: Tracking relationships with pointers

- Pointing to mutable object.
  - It's changing as you look at it.
- Pointers are ephemeral and out of your control.
  - Pointers are different each time you run.
- There could be two Student objects with the same Student ID.
  - Same real person, different objects.

Takeaways

- Model relationships with a *value*
  - Something immutable
  - With equality semantics
  - That represents identity

Let's model the registration process

- Construct a University
  - Construct a Student
  - Construct the Course they want
- Call the register method on Student with Course
- ...
- Profit!

LIES!

Lie #4: We want to model the actors

- Why is the register method on Student?
  - Because "A student registers for a course"
  - We need to record the course in Student.courses
- Why is the register method on Course?
  - Because we need to record the student in Courses.students.
- Will these lists get out of sync?
- Why are we modeling students and courses, anyway?
  - OO programming was made for *Simulations*.

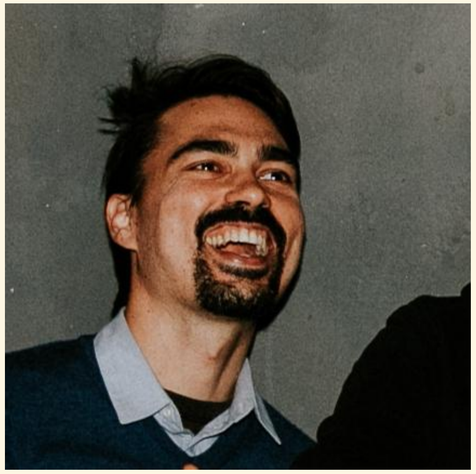
Takeaways

- What should we simulate?
- Exercise:
  - How would you run registration if all you had were paper and pen?

LIES!

Object Oriented Programming

- You can do all of this with OOP.
  - Many good programmers already do.
- Lie #1: Noun/verb
  - Many OO refactorings convert between class and method
- Lie #2: Queries are interesting
  - Simulation is great for counterfactuals.
- Lie #3: Relationships with pointers
  - Value objects.
- Lie #4: Modeling all of the actors
  - Easily done in OOP.
- OOP is great for exploring different models.



Eric Normand

LispCast

Follow Eric on:

Eric Normand

@EricNormand

lispcast.com

eric@lispcast.com