CS4450/7450: Introduction

Professor William L. Harrison

September 10, 2018

CS4450/7450: Principles of Programming Languages

- What is this course about?
  - Programming language Design

CS4450/7450: Principles of Programming Languages

- What is this course about?
  - Programming language Design
  - Data abstraction, Language Classes, Types
  - Explore these ideas by writing interpreters in Haskell

CS4450/7450: Principles of Programming Languages

- What is this course about?
  - Programming language Design
  - Data abstraction, Language Classes, Types
  - Explore these ideas by writing interpreters in Haskell
- Administrivia
  - Grading
  - Texts

CS4450/7450: Principles of Programming Languages

- What is this course about?
  - Programming language Design
  - Data abstraction, Language Classes, Types
  - Explore these ideas by writing interpreters in Haskell
- Administrivia
  - Grading
  - Texts
- Course Webpage
  - TBD

Grading

- Midterm (Friday, 10/5): 25%
- Written Homeworks + Programming: 40%
- Final: 35%

(Regrade Policy)

Requests for re-grades must be made in writing within 7 days of receiving graded HW/test. There will be absolutely no exceptions.

Personnel

- Instructor: Professor William L. Harrison
  Office: 318 EBN
  Office hours: *By appointment; email me first*
- Teaching Assistants:
  - Trevor Bajkowski
  - Matthew Deardorff

Textbooks

Both texts are online:

- *Learn You a Haskell for Good* by Miran Lipovaca
  Highly amusing and informative; available online.
- *Anatomy of Programming Languages* by William Cook.
- Slides, of course.

The following are good but not required.

- *Programming in Haskell* by Graham Hutton
- *Thinking Functionally with Haskell*, Richard Bird, Cambridge University Press.

What is this course about?

Why would you study interpreters and compilers?

What is this course about?

Why would you study interpreters and compilers?

- Writing interpreters/compilers involves a lot of technical skills that you need to use together.
    - Writing interpreter/compiler helps improve those skills to become a better software developer in general.

What is this course about?

Why would you study interpreters and compilers?

- Writing interpreters/compilers involves a lot of technical skills that you need to use together.
    - Writing interpreter/compiler helps improve those skills to become a better software developer in general.
- You really want to know how computers work.
    - Often interpreters and compilers look like magic. And you shouldn't be comfortable with that magic.
    - You want to demystify the process of building an interpreter and a compiler, understand how they work, and get in control of things.

What is this course about?

Why would you study interpreters and compilers?

- Writing interpreters/compilers involves a lot of technical skills that you need to use together.
    - Writing interpreter/compiler helps improve those skills to become a better software developer in general.
- You really want to know how computers work.
    - Often interpreters and compilers look like magic. And you shouldn't be comfortable with that magic.
    - You want to demystify the process of building an interpreter and a compiler, understand how they work, and get in control of things.
- You want to create your own programming language or domain specific language.
    - If you create one, you will also need to create either an interpreter or a compiler for it.
    - Recently, there has been a resurgence of interest in new languages: Elixir, Go, Rust just to name a few.

Academic Honesty

- Your work must be your own
    - Discussion is fine and encouraged.

Academic Honesty

- Your work must be your own
  - Discussion is fine and encouraged.
  - “Discussion” means talking with your mouth or chalkboard only.
  - Realize that if you email someone code as an explanation, they may cut and paste it into something they turn in.

Academic Honesty

- Your work must be your own
  - Discussion is fine and encouraged.
  - “Discussion” means talking with your mouth or chalkboard only.
  - Realize that if you email someone code as an explanation, they may cut and paste it into something they turn in.
- Consequences of Academic Dishonesty
  - 1st offense: 0 on offending assignment & warning
  - 2nd offense: Automatic F grade; case forwarded to Provost.

Academic Honesty

- Your work must be your own
  - Discussion is fine and encouraged.
  - “Discussion” means talking with your mouth or chalkboard only.
  - Realize that if you email someone code as an explanation, they may cut and paste it into something they turn in.
- Consequences of Academic Dishonesty
  - 1st offense: 0 on offending assignment & warning
  - 2nd offense: Automatic F grade; case forwarded to Provost.
  - No exceptions!

Academic Honesty

- Your work must be your own
  - Discussion is fine and encouraged.
  - “Discussion” means talking with your mouth or chalkboard only.
  - Realize that if you email someone code as an explanation, they may cut and paste it into something they turn in.
- Consequences of Academic Dishonesty
  - 1st offense: 0 on offending assignment & warning
  - 2nd offense: Automatic F grade; case forwarded to Provost.
  - No exceptions!
- Your continued enrollment in this class implies your consent to these rules.

Email Use & Abuse

- I will only answer emails from 9am-5pm Monday-Friday
    - ...actual emergencies excepted.

Email Use & Abuse

- I will only answer emails from 9am-5pm Monday-Friday
    - ...actual emergencies excepted.
- It is completely at my discretion whether I answer your email.

Email Use & Abuse

- I will only answer emails from 9am-5pm Monday-Friday
  - ...actual emergencies excepted.
- It is completely at my discretion whether I answer your email.
- Rude, snarky, obnoxious, and/or threatening emails will not be answered and, if appropriate, will be forwarded on to my department chairman and/or provost.

How to get an A in CS4450/7450

- When assigned reading, read it carefully.
    - Don't skim, *read*!

How to get an A in CS4450/7450

- When assigned reading, read it carefully.
    - Don't skim, *read*!
- When given a programming assignment, start early.
    - Last minute, start-the-night-before may work in other classes, but it's a prescription for disaster in CS4450.

How to get an A in CS4450/7450

- When assigned reading, read it carefully.
    - Don't skim, *read*!
- When given a programming assignment, start early.
    - Last minute, start-the-night-before may work in other classes, but it's a prescription for disaster in CS4450.
- Realize that you alone are responsible for your performance; be disciplined.

How not to get an A in CS4450

Case Study 1

Professor,
While you were away, I attempted to inquire about my grade via email and your auto email responded saying you were out of the country. I was inquiring about my grade because as a Senior slated to graduate in the fall I was hoping to earn an A- in your class after we had taken the final. I needed the A- for gpa purposes for job applications that I had to begun to fill out for when I graduate in the fall. I work in Dr. Kazic's Lab and she always speaks highly of you so I was hopeful that there may have either been a miscalculation on my final or class attendance points or a way for me to earn a few more points so that I may raise my grade and graduate in the fall.
Is there any way for me to raise my grade a degree?
Gratitude, ...

How not to get an A in CS4450

Case Study 2

I have more than three days left.
The final isn't till next Thursday.
I'm not asking for pity, I'm asking that you do help me with specific material and not throw me to your TA's. I talked to you about this before Thanksgiving, but was out of town all last week for an interview with Amazon. I wasn't able to come in then so there was no point in talking to you about it. I've really been trying to keep up, but I just haven't been able to.

I'll come in multiple times, and only ask specific questions. I'm not just going to come in and say "teach me".

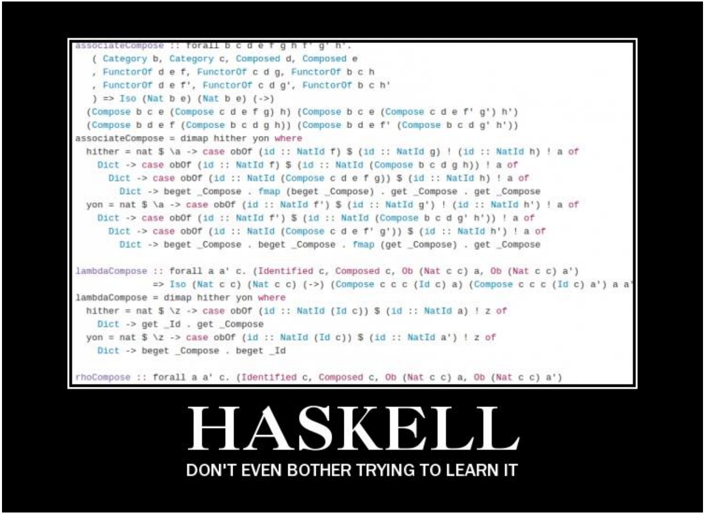
How not to get an A in CS4450

Case Study 2, cont'd

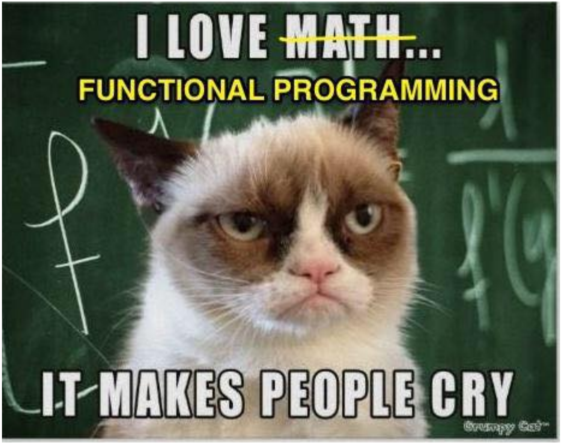
Quite frankly, I'm disappointed that this is the response I get when I ask for help. You still haven't given me potential times to come in, and I assume that by including your TA's I'm going to get the typical response of "go talk to your TA's". I don't pay this University though to talk to TA's. I pay to get help from a professor.

Also, you are the third professor that has been stubborn to work with me with traveling this semester. I'm not taking vacations, I'm trying to get a job at some of the biggest tech companies in the US (Google, Amazon, and Microsoft). This University is supposed to enable me to succeed, not impede me.

So, I'll ask again. When can I come talk to you about the material in class so I can pass this class, and more importantly, understand the material?



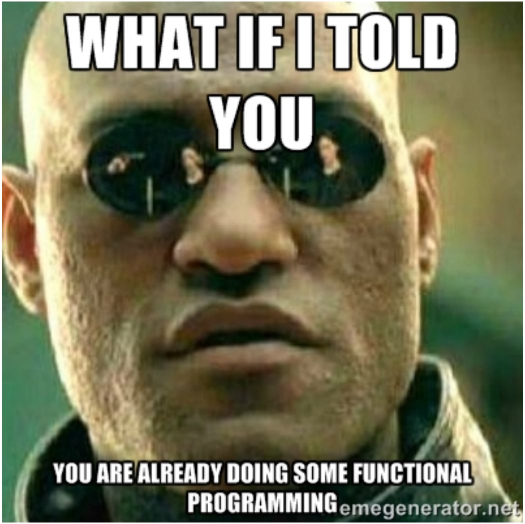
```
associateCompose :: forall b c d e f g h f' g' h'.
   ( Category b, Category c, Composed d, Composed e
   , FunctorOf d e f, FunctorOf c d g, FunctorOf b c h
   , FunctorOf d e f', FunctorOf c d g', FunctorOf b c h'
   ) => Iso (Nat b e) (Nat b e) (->)
  (Compose b c e (Compose c d e f g) h) (Compose b c e (Compose c d e f' g') h')
  (Compose b d e f (Compose b c d g h)) (Compose b d e f' (Compose b c d g' h'))
associateCompose = dimap hither yon where
  hither = nat $ \a -> case obOf (id :: NatId f) $ (id :: NatId g) ! (id :: NatId h) ! a of
    Dict -> case obOf (id :: NatId f) $ (id :: NatId (Compose b c d g h)) ! a of
      Dict -> case obOf (id :: NatId (Compose c d e f g)) $ (id :: NatId h) ! a of
        Dict -> beget _Compose . fmap (beget _Compose) . get _Compose . get _Compose
  yon = nat $ \a -> case obOf (id :: NatId f') $ (id :: NatId g') ! (id :: NatId h') ! a of
    Dict -> case obOf (id :: NatId f') $ (id :: NatId (Compose b c d g' h')) ! a of
      Dict -> case obOf (id :: NatId (Compose c d e f' g')) $ (id :: NatId h') ! a of
        Dict -> beget _Compose . beget _Compose . fmap (get _Compose) . get _Compose

lambdaCompose :: forall a a' c. (Identified c, Composed c, Ob (Nat c c) a, Ob (Nat c c) a')
              => Iso (Nat c c) (Nat c c) (->) (Compose c c c (Id c) a) (Compose c c c (Id c) a') a a'
lambdaCompose = dimap hither yon where
  hither = nat $ \z -> case obOf (id :: NatId (Id c)) $ (id :: NatId a) ! z of
    Dict -> get _Id . get _Compose
  yon = nat $ \z -> case obOf (id :: NatId (Id c)) $ (id :: NatId a') ! z of
    Dict -> beget _Compose . beget _Id

rhoCompose :: forall a a' c. (Identified c, Composed c, Ob (Nat c c) a, Ob (Nat c c) a')
```

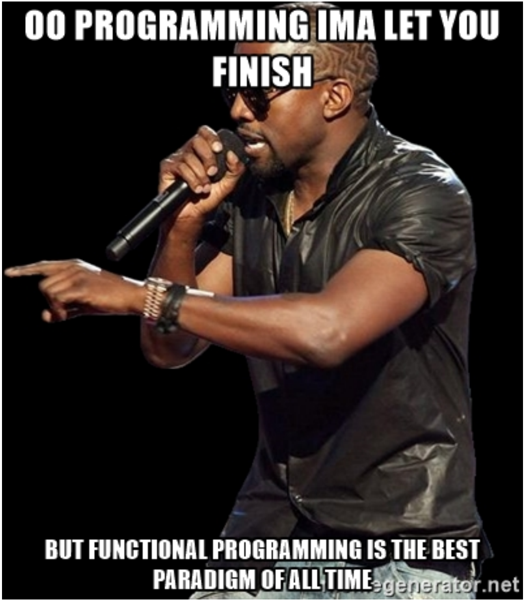

I LOVE ~~MATH~~...
FUNCTIONAL PROGRAMMING
IT MAKES PEOPLE CRY
Grumpy Cat™

AND THIS IS HOW
WE WRITE HELLO WORLD
IN HASKELL

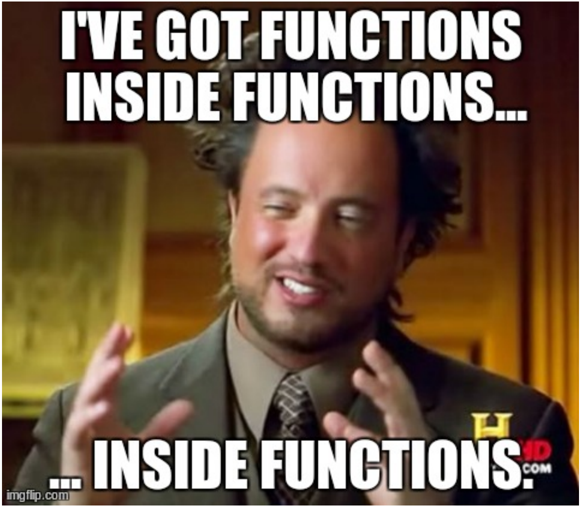
Actually, it's just:

```
main = print "Hello World"
```


WHAT IF I TOLD YOU
YOU ARE ALREADY DOING SOME FUNCTIONAL PROGRAMMING
emegenerator.net


OO PROGRAMMING IMA LET YOU FINISH
BUT FUNCTIONAL PROGRAMMING IS THE BEST PARADIGM OF ALL TIME
egenerator.net

HEY GIRL.
LIKE ALL VALUES IN HASKELL, MY LOVE FOR YOU IS IMMUTABLE.


I'VE GOT FUNCTIONS INSIDE FUNCTIONS...
... INSIDE FUNCTIONS.
imgflip.com

omputer
usiness
Center
DO I LOOK LIKE I KNOW
WHAT A MONAD IS?
made on imgur