CSCI 334: Principles of Programming Languages

Lecture 10: Functional Programming

Instructor: Dan Barowy Williams

Announcements

Midterm exam next class

Announcements

Midterm exam *next class*

You should have feedback for all HW if not, please let me know!

"Recursive Functions []" (McCarthy)		
Lisp	<u>C</u>	







"Recursive Functions []" (McCarthy)			
	Lisp	<u>C</u>	
	car	head	
	cdr	tail	

"Recursive Functions [...]" (McCarthy)

Lisp	<u>C</u>
car	head
cdr	tail

cons

"Recursive Functions [...]" (McCarthy) Lisp <u>C</u> car head cdr tail cons prepend

























"Growth" mindset

"In a fixed mindset students believe their basic abilities, their intelligence, their talents, are just fixed traits. They have a certain amount and that's that, and then their goal becomes to look smart all the time and never look dumb. In a growth mindset students understand that their talents and abilities can be developed through effort, good teaching and persistence."

— Carol Dweck (Lewis and Virginia Eaton Professor of Psychology at Stanford University)

Mental technique #4

"Growth" mindset

"In a fixed mindset students believe their basic abilities, their intelligence, their talents, are just fixed traits. They have a certain amount and that's that, and then their goal becomes to look smart all the time and never look dumb. In a growth mindset students understand that their talents and abilities can be developed through effort, good teaching and persistence."

— Carol Dweck (Lewis and Virginia Eaton Professor of Psychology at Stanford University)

Individuals with a "growth" mindset are more likely to continue working hard and succeed—despite setbacks.

Mental technique #4

"Growth" mindset

Mental technique #4

"Growth" mindset

Your brain is a machine designed to accommodate to a changing world.



Demonstration (again)

Mental technique #4

Demonstration (again)

Demonstration (again)

If that made sense to you, raise your hand.

Mental technique #4

Demonstration (ungarbled)

Mental technique #4

Demonstration (ungarbled)

Mental technique #4

Demonstration



Demonstration (again)

Mental technique #4

Demonstration (again)

Anil Seth, "Your brain hallucinates your conscious reality"

Why am I telling you this?



Why am I telling you this?



This course is about priming your brain with different ways of thinking about programming.

Why am I telling you this?

Why am I telling you this?

You can be a programmer without these ideas.

Why am I telling you this?

You can be a programmer without these ideas.

But make the effort to internalize these concepts and you will see their application everywhere.

Why am I telling you this?

You can be a programmer without these ideas.

But make the effort to internalize these concepts and you will see their application everywhere.

You will be a *clearer* thinker and a *better* programmer.

Three amazing concepts from FP

Three amazing concepts from FP

First-class functions

Three amazing concepts from FP

- First-class functions
- Higher-order functions

Three amazing concepts from FP

- First-class functions
- Higher-order functions

•map

Three amazing concepts from FP

- First-class functions
- Higher-order functions
- map
- fold









"first class" function

Functions are values in a programming language





















































fold right

(reduce **#'+ '**(1 2 3):initial-value 0













what does this print?

(reduce **#**'append '((1) (8))

what does this print?

(reduce **#**'append '((1) (8))

:initial-value '(w i l l i a m s))



how about?

(reduce **#**'append '((1) (8))

how about?

(reduce **#**'append '((1) (8))

:initial-value '(w i l l i a m s)

how about?

(reduce **#**'append '((1) (8))

:initial-value '(w i l l i a m s)

:from-end t)

fold

fold

structural recursion \rightarrow fold it!

fold

structural recursion → fold it!
(in a nutshell: any problem that recurses on a subset of input)

fold

 $structural\ recursion \rightarrow {\rm fold\ it!}$ (in a nutshell: any problem that recurses on a subset of input)



list length

fold

structural recursion \rightarrow fold it! (in a nutshell: any problem that recurses on a subset of input)

• • • Ø



list length

tree height

