3-2) Functor Quiz

Due No due date **Points** 4 **Questions** 4
Available Sep 10 at 12am - Sep 10 at 11:59pm 1 day **Time Limit** None

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	2 minutes	0 out of 4

Score for this quiz: **0** out of 4
Submitted Sep 10 at 1:01pm
This attempt took 2 minutes.

Question 1 — 0 / 1 pts

My code needs to use an ADT to track all my students in alphabetical order. What would be the best pick?

ou Answered
- (•) List

- ( ) Vector

orrect Answer
- ( ) Set

- ( ) Map

A Set's power is in keeping things in order. If I didn't want order and just wanted to hold the students, I would pick List since it has faster adds and iterators.

Question 2

0 / 1 pts

How about to keep track of how much money each Student owes the school?

- List
- **ou Answered** Vector
- Set
- **orrect Answer** Map

Map is the only one that can associate two kinds of data with each other.

Question 3

0 / 1 pts

I have a struct called Square. What should I write to make Squares sortable by area?

operator <

ou Answered

operator ()

BySize

orrect Answer

Square::BySize

We don't want < because sorting by edge length and sorting by area both seem useful. BySize by itself would be the C way of passing a function pointer. We want to make scoped functors.

Question 4

0 / 1 pts

I have a struct called Line I want to sort by length. Same question

orrect Answer

operator <

ou Answered

operator ()

ByLength

Line::ByLength

This is an important distinction I wanted to make to show I'm not totally anti-operator overloading. If a class or struct has only one property, or is clearly a mathematical construct, then < actually does make sense. It's more readable for math to look like math.

Quiz Score: **0** out of 4