4-1) Vector Quiz

Due No due date **Points** 5 **Questions** 5 **Time Limit** None

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	1 minute	3 out of 5

Score for this quiz: **3** out of 5
Submitted Sep 15 at 12:57pm
This attempt took 1 minute.

Question 1 — 1 / 1 pts

What is the point of writing the Big 3?

- Don't leak memory
- Don't get a pointer to another object's private data
- Don't crash
- All of the above

Correct!

All of the trouble comes from how a default copy does nothing but copy the value of a variable. Pointers are just numbers. I don't want my pointer to point at your memory.

Question 2

0 / 1 pts

An ADT can hold pointers but not objects

ou Answered

- (•) True

orrect Answer

- ( ) False

I'm going to recommend using pointers with containers because the lifetime of objects gets confusing otherwise. When you add to a container, it takes a copy. When you get from a container it gives you a reference to the object inside.

If you are using pointers, those are the same. With objects though... where did the first object that was in the push_back go?

Question 3

1 / 1 pts

To do my Vector homework, I am just going to finish vector.h and turn it in.

- ( ) True

Correct!

- (•) False

You are going to debug and test it too. This is especially crucial with template classes, as any method you don't test gets deleted.

Question 4

0 / 1 pts

I'd use a vector if my only concern was accessing data at random as fast as possible.

orrect Answer

- True

ou Answered

- False

Other containers have fast lookups (unordered_map) but if you don't need the overhead of maps and you know where you want to put your data, Vector is faster.

Other considerations mean picking different ADTs. If I need to delete at random I wouldn't pick Vector since it is O(n) at removing.

I'm brainstorming off topic because programming is fun, but you could take a u_m, make the key 0-n like vector with values as your data. You'd end up with a lot of empty spots in your vector though. Sorry, where was I?

Question 5

1 / 1 pts

The difference between stack memory and heap memory is that stack is known at compile time and heap is at run time.

Correct!

- (•) True
- ( ) False

Yup. If you peeked in to an exe, you'd see all of the code and space for all of the variables are there.

Quiz Score: **3** out of 5