

Homework 16: Move Semantics, Value Categories, and Smart Pointers

Instructor: Mehmet Emre

CS 32 Spring '22

Due: 6/1 12:30pm

Name & Perm #:

Homework buddy (leave blank if you worked alone):

Reading: The relevant resources mentioned in lecture 14.

1

In class, we explored what `std::move` does, and did some experiments to show that `std::move(x)` doesn't actually move `x` anywhere. What is the semantics of `std::move` (that is, what does `std::move` actually do)?

2 2 pts

In the `UniquePtr` class we developed in class, how we changed the copy and move constructors as well as the assignment operators to guarantee that only 1 `UniquePtr` can point to an object at a time.

How do the move constructor and the move assignment operator provide this guarantee?

3

In the class, I argued that `std::unique_ptr` is a good first choice when needing a pointer.

1. (1 pt) What is the advantage of `std::unique_ptr` over a plain pointer (like `T *`)?
2. (2 pt) What are the two advantages of using `std::unique_ptr` over `std::shared_ptr`?

4

Reference counting cannot detect that certain objects are not reachable (and can be safely destroyed).

1. (2 pt) What causes this limitation (how the objects should be used for reference counting to fail to determine that they are unreachable)?
2. (1 pt) What machinery is provided by the C++ standard library to get around this limitation?