This interface for the container class is a complete interface: walls and mirrors refers to a complete interface as “A complete interface for a class is one that will allow programmers to accomplish any reasonable task, given the responsibilities of that class”(12). Which I would argue is a very fitting name for the interface that is used in the container class, and here is why. The container class in particular contains functions that allow for them to view everything inside of the container, calculate the total tonnage, along with checking to see if things are able to even be added to it. This would all fall under the presumption of a complete interface, however there are classes which by themselves could be argued to be a minimal interface which is described as: “...is one that contains a method if and only if that method is essential to that class’s responsibilities.”(12). While it could be argued that the shippable class fits that requirement, the class simply just keeps the variables and has mutators and assessors without it doing much else which is the only thing essential to the class responsibilities.

Carrano, Frank M., and Timothy Henry. *Data Abstraction & Problem Solving with C++: Walls and Mirrors*. Pearson, 2017.