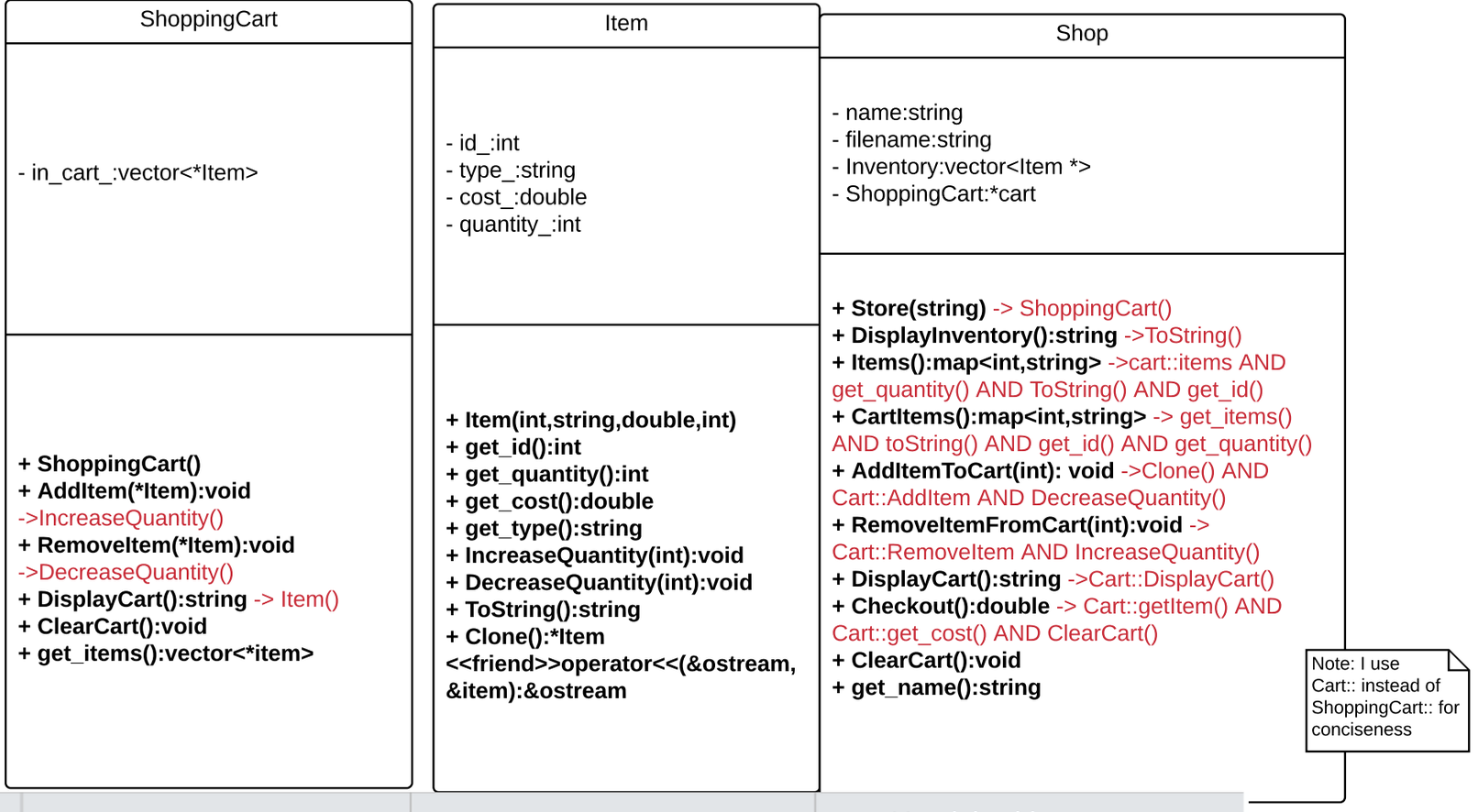
Part 2:

Clone is used when adding an item to cart, it creates a pointer to said item that is added to the item pointer vector in cart. I think it is included in the item definition because it accesses many of item's private attributes.



cohesive (one single abstraction)	ShoppingCart itself is very cohesive	Yes, contains everything item needs	Yes, it is able to access everything it needs to
complete (provides a complete interface)	No, many of the methods shop uses should have been used by ShoppingCart.	Yes, item does not need to do much but exist and hold variables, and clone itself	Yes, as a user, it is mostly clear how it interacts with program
clear (the interface makes sense)	Yes, I look at a function name and know what it does	As stated above, very clear what it does	No, besides having names that sound similar to ShoppingCart, it uses methods inside shoppingCart often
convenient (makes things simpler in the long run)	Absolutely Not, so many functions in shop should have been implemented in cart instead	Yes, allows us to have cart make a vector of items	No , functions like AddItemToCart existing in Shop makes no sense
consistent (names, parameters, ordering, behavior should be consistent)	No, constructor wasn't even defined in .h file	Yes, naming conventions seem consistent	No, ideally cart and store should have different .cpp files, or the order of function defs should at least be consistent with their .h files