9. Constructors of whole classes and parent classes

- Which classes are aggregates of other classes? Checking all constructors of whole classes if they initialize for their parts?

Aggregates:

- Store aggregates Media.

- Cart aggregates Media.

- CompactDisc aggregates Track.

Store Class

- Constructor: Initializes the list of Media.

Cart Class

- Constructor: Initializes the list of Media.

Disc Class

- Constructor: Sets properties for Disc, and indirectly via inheritance, initializes Media attributes.

DigitalVideoDisc Class

- Constructor: Sets properties specific to DigitalVideoDisc and initializes inherited ones.

CompactDisc Class

- Constructor: Likely initializes the List<Track>.

Track Class

- Constructor: Initializes these properties.

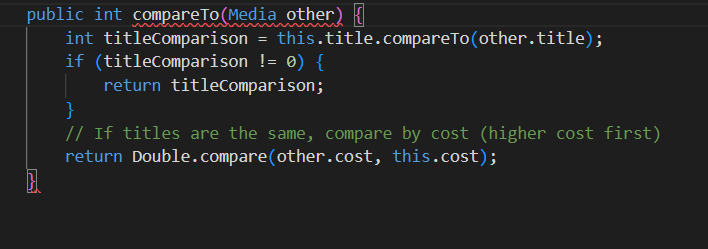
10. If the passing object is not an instance of Media, what happens?

- Return False

12. Sort media in the cart

Question: Alternatively, to compare items in the cart, instead of using Comparator, we can use the Comparable interface and override the compareTo()method. You can refer to the Java docs to see the information of this interface.

Suppose we are taking this Comparable interface approach.

- In those classes, how should you implement the compareTo()method be to reflect the ordering that we want?

- Can we have two ordering rules of the item (by title then cost and by cost then title) if we use this Comparable interface approach?

No because the Comparable interface allows only one natural ordering for a class. If we need multiple ordering rules (like sorting by title or cost), we must replace with Comparator instead.

- Suppose the DVDs has a different ordering rule from the other media types, that is by title, then decreasing length, then cost. How would you modify your code to allow this?

As DVDs has different ordering rule from other media types, We can override compareTo() method in DigitalVideoDisc class. After that, DVDs will be compare by the overridden compareTo() method.

Compare test:

