Comp 303 A2

I used the provided baseline code but made some changes to it.
There are three watchable objects: Movie, Episode, and TVShow. All the watchable objects are relevant to the same operations: being added to a watchlist, being watched and removed from the watchlist. Therefore I changed WatchList to a generic class. It can be either a wactchlist of movies, episodes or TVshows.
A movie that has sequels and all episodes from a TVshow are comparable. So I create an interface Comparable. The method comparesTo defines an order between two movies or two episodes.
TVShow and WatchList are bingeable objects. The interface Bingeable implements Iterable, so that we can define specific iterator in TVShow and WatchList to access all the elements.
The client can generate a watchlist given his own algorithm. For this part I create an interface Feature as a function object. The client can define the algorithm by creating a Feature class and overriding toAdd(). The method returns a list of elements that the client wants to add into the watchlist. In the class Library, the method List_generation() will take Feature as input and return a watchlist.