ASSIGNMENT 1 EXTENTION

Implementing WatchList class:

- For the extension of the assignment, I implement Question 8 of the assignment which is the class WatchList. I followed the specifications of the class written in coderunner and used my passed unit tests to test the implementation. I include the classes that I made previously from Question 1 – Question 4 (Director, Genre, Actor, Movie classes) into this question.

- Constructor `def __init__(self)` is created without any parameter (only self). Inside the constructor, I create an attribute self.watch_list with the type "list" which will be used throughout the class WatchList. Also, a function `def size(self)` is made to get the size/length of the self.watch_list list which will be used later.

- `def add_movie(self, movie)` passes the Object Class Movie to be added into the end of self.watch_list using .append() method. Note that the new Movie object is only added if it's not yet in self.watch_list, if it has been added into the list before (the new Movie Object is the same as one of the Movie Object within the list), then the method of adding into the list is not executed. There's no duplicate Movie Object inside the list.

- `def remove_movie(self, movie)` passes the parameter Movie object. This function searches whether the movie is present in the self.watch_list list by using the .index(movie) method to get the index if the movie is inside the list, if a match is found, then remove it from the list using .pop(index) method. I used try and except method where if it gives a "ValueError", then it means that there's no such object in the list, which will then "pass" this method and didn't do anything to the list.

- `def select_movie_to_watch(self, index)` allows user to enter an index value as parameter. It chooses the movie in that particular index from self.watch_list to be watched. Before choosing the movie in the given index, `self.size()` I called which gives the length of the list self.watch_list and store it in the variable lst_len. If the index passed in the parameter is greater than or equal to lst_len, it indicates that the index is out of bounds and getting the movie from that index is impossible, which will return None. If the index is within the range of the length of self.watch_list, then the function will return the movie in that index.

- `def select_movie_to_watch(self)` gets the first movie within the list self.watch_list. First it checks the length of the list self.watch_list, if the length is 0, then there's no movie in the list which will return None. If the length of the list is greater than or equal to 1, it will return self.watch_list[0] which is the first movie in the list.

- `def __iter__(self)` and `def __next__(self)` iterates self.watch_list starting from the first index self.watch_list[0] all the way to the end of the list by incrementing the index by 1. It raises StopIteration() once index is out of bounds.

Unit Tests (from Question 8 in coderunner):

watchlist = WatchList()
print(f"Size of watchlist: {watchlist.size()}")
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
print(watchlist.first_movie_in_watchlist())
	Size of watchlist: 0
	<Movie Moana, 2016>

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
print(f"Size of watchlist: {watchlist.size()}")
	Size of watchlist: 3

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Moana", 2016))
print(f"Size of watchlist: {watchlist.size()}")
	Size of watchlist: 1

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
print(f"Size of watchlist: {watchlist.size()}")
watchlist.remove_movie(Movie("Ice Age", 2002))
print(f"Size of watchlist: {watchlist.size()}")
	Size of watchlist: 3
	Size of watchlist: 2

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
print(f"Size of watchlist: {watchlist.size()}")
watchlist.remove_movie(Movie("Guardians of the Galaxy", 2012))
print(f"Size of watchlist: {watchlist.size()}")
	Size of watchlist: 2
	Size of watchlist: 2

watchlist = WatchList()
print(f"Size of watchlist: {watchlist.size()}")
print(watchlist.first_movie_in_watchlist())
	Size of watchlist: 0
	None

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
print(f"Size of watchlist: {watchlist.size()}")
watchlist.select_movie_to_watch(1)
print(watchlist.select_movie_to_watch(1))
	Size of watchlist: 3
	<Movie Ice Age, 2002>

GITHUB LINK : https://github.com/NatRivers/tlan121---CS235---/tree/master/A1

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
print(f"Size of watchlist: {watchlist.size()}")
watchlist.select_movie_to_watch(1)
print(watchlist.select_movie_to_watch(5))
	Size of watchlist: 3
	None

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
iterate = iter(watchlist)
print(next(iterate))
	Movie Moana, 2016>

watchlist = WatchList()
watchlist.add_movie(Movie("Moana", 2016))
watchlist.add_movie(Movie("Ice Age", 2002))
watchlist.add_movie(Movie("Guardians of the Galaxy", 2012))
iterate = iter(watchlist)
print(next(iterate))
print(next(iterate))
print(next(iterate))
 	
print(next(iterate, '-1'))
	<Movie Moana, 2016>
	<Movie Ice Age, 2002>
	<Movie Guardians of the Galaxy, 2012>
	-1

GITHUB LINK : https://github.com/NatRivers/tlan121---CS235---/tree/master/A1