ECE 218
Lab #4
Linked Lists – A Movie Repository I: The Movies and Ratings

In this assignment, you are to implement a repository for movies as the first part of an online movie streaming service. This first part will implement the list of Movies and the Ratings of each movie, each as single-link linked lists.

You need to provide a report that shows: **(10)**

1. Your class designs
2. The algorithms used adding/deleting movies and reviews
    a. This should be in pseudo-code or flowcharts, not just the code you ended up writing
3. Compiling instructions
4. Sample runs (screenshots are fine)

Part1a: A Rating Class (5)

1. Design a class, *Rating*, to hold the rating information.
    a. Private:
        i. Should have at least: rating (int), comment (string)
    b. Public:
        i. Constructor(s)/Destructor as needed
        ii. Setters to set the data
        iii. Method to print the object[i]
2. Test these to make sure this works.

Part1b: A RatingList Class (20)

1. Design a class, *RatingList*, which implements a single-link linked list of Ratings.
    a. A RatingNode struct
        i. Private:
            1. rating (Rating), *next (RatingNode)
    b. Private:
        i. Should have at least: *front (RatingNode)
    c. Public:
        i. Constructor(s)/Destructor
        ii. addRating(Rating &r) method
            1. adds the Rating r at the end of the list
        iii. deleteRating(Rating &r)
            1. finds and deletes the rating r from the list
        iv. Method to print the entire list[i]
2. Test to make sure this works

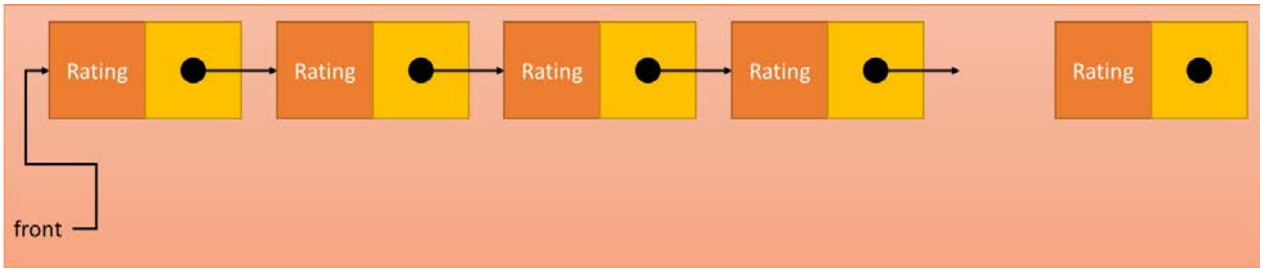

Figure 1 RatingList Linked List

Part 2a: A RTime Class (5)

1. Design a class, *RTime*, to hold time information. This will be used to hold the running time of a movie.
    a. Private:
        i. Should have at least: hours (int), minutes (int)
    b. Public:
        i. Constructor(s)/Destructor as needed
        ii. Setters to set the data
        iii. Method to print the object[i]
2. Test this to make sure this works.

Part 2b: A Movie Class (10)

1. Design a class, *Movie*, to hold movie information.
    a. Private:
        i. Should have at least: title (string), runtime (RTime), *ratings (RatingsList)
    b. Public:
        i. Constructor(s)/Destructor as needed
        ii. Setters to set the data
        iii. addRating(Rating &r) method to add a rating
        iv. deleteRating(Rating &r) method to delete a rating
        v. deleteAllRatings() method to delete all ratings
        vi. compare(Movie *m) compare current with m, -1=less than, 0=equal. 1=greater than
        vii. Method to print the object[i]
2. Test this to make sure this works.

Part2c: A MovieList Class (35)

1. Design a class, *MovieList*, which implements a single-link linked list of Movies sorted by title.
    a. A MovieNode struct
        i. Private:
            1. *movie (Movie), *next (MovieNode)
    b. Private:
        i. Should have at least: *front (MovieNode)
    c. Public:
        i. Constructor(s)/Destructor
        ii. addMovie(Movie *m) adds a movie to the list sorted by title
        iii. removeMovie(string t) removes movie with title t
        iv. deleteAllMovies() removes all movies

v. addRating(string t, Rating &r) method
    1. adds the Rating r to the movie of title t
vi. deleteRating(string t, Rating &r)
    1. finds and deletes the rating r from the movie title t
vii. printMovie(string t) print movie and ratings with title t
viii. Method to print the entire list[i]

2. Test to make sure this works

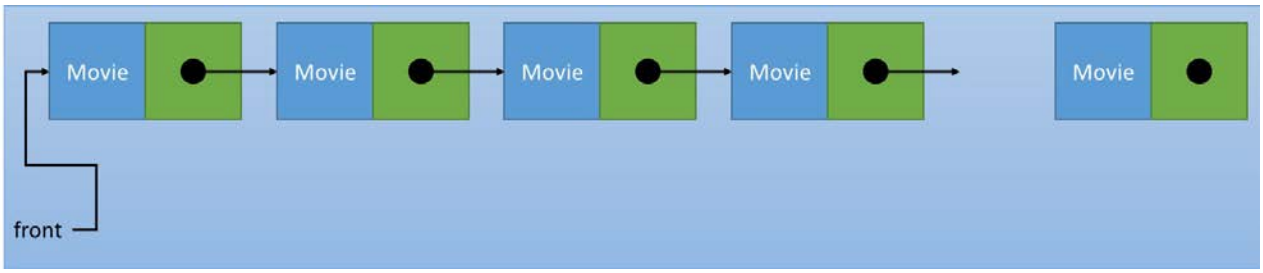

Figure 2 MovieList Linked List

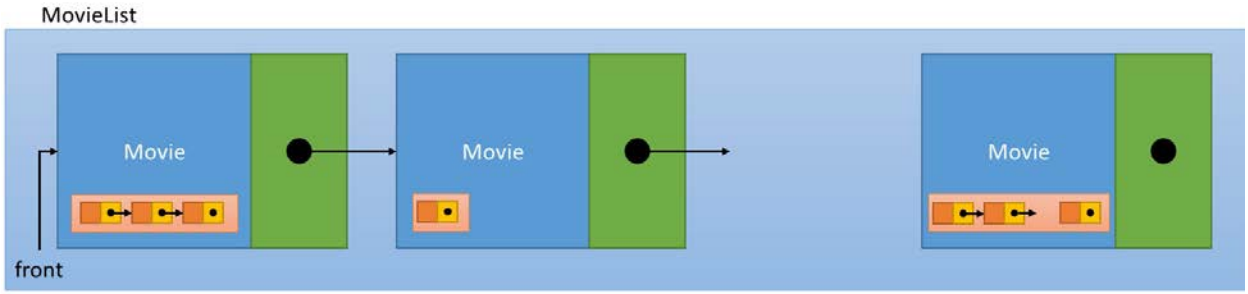

Figure 3 Detail of MovieList (shows separate RatingList in each Movie)

Part 3: A MovieNet Class (15)

1. Design a class, *MovieNet*, which acts as the entry for our online movie streaming system
    a. Private:
        i. name (string), *movies (MovieList)
    b. Public:
        i. Constructor(s)/Destructor as needed
        ii. loadData(string file) loads the movie list from a file
        iii. storeData(string file) stores the movie list to a file
        iv. Print method to print out the object[i].

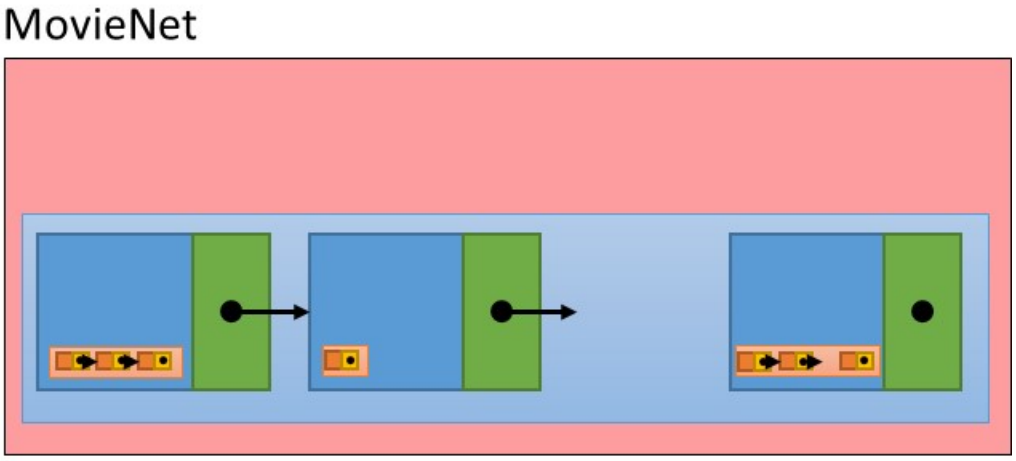

Figure 4 MovieNet detail structure showing MovieList inside

File Format:

START
Movie title
Runtime (in minutes)
RSTART
Rating
Comment
Rating
Comment
..
REND
Movie title
Runtime (in minutes)
RSTART
Rating
Comment
Rating
Comment
..
REND
….
END

[i] All print functions should allow for specification of the output stream and output in a format compatible with the File Format described at the end of the document.