CPSC 233 – Coding Challenge 2 – Practice 2

This is a practice coding challenge. You may ask other students and your TA any questions you wish. Note that you are expected to complete the actual coding challenge independently. Having a solution for this coding challenge will not help you come up with your own solution for the actual coding challenge. Only if you can solve this practice independently can you be confident that you can complete the actual coding challenge successfully. See previous coding challenges for rules for coding challenges and best practices for success.

Requirements

Remember to always create a 'skeleton' first and compile and test this using the provided JUnit test. For this coding challenge, you will be asked to create two classes. Make sure you create a skeleton of both classes first. WebCAT needs a version of both classes that compile with the tests placed in a zip file. If one of the classes is missing or does not compile, none of the tests will be run and all tests will be marked as failed.

Requirements for Skeleton of Movie and Festival

	Movie	Festival
Instance Variables	title: String rating: int	name: String movieList: ArrayList<Movie>
Constructors	1: arguments: title, rating 2: copy constructor	1. arguments: name
Getters	title and rating	name and movieList
Setters	title and rating	
Methods	getCategory(): void	addMovie(Movie): void getMovieWithLowestRating():Movie

Additional Requirements for Movie Class

- *Title*: The instance should always store the string in all upper case, if the method is called with a string that contains any lower case characters, the lower case characters should be converted to upper case.
- *Rating* must be a value between 0 and 10 (inclusive). If the rating provided is outside of that range, the rating should remain unchanged.
- *getCategory*: A movie is considered A category if its rating is 9 or 10, B category if the rating is 7 or 8, C category if the rating is 5 or 6, D category if the rating is 3 or 4 and F category otherwise.

Additional Requirements for Festival Class

- All instance variables must be completely encapsulated. There should be no privacy leaks.
- *getMovieWithLowestRating* returns the movie in movieList that has the lowest rating. If the list is empty, this method will return null.
- Don't forget to import java.util.ArrayList