M1A - 52 Creative Assignment

1. Program requirements (20 points): You must include detailed program requirements describing a realistic real-world problem related to this semester topic area (see welcome module). The requirements must include the use of a BAG for storing some data. You must include at least one paragraph with 7 or more lines.

This program models a customer rating system for an autonomous vehicle, (also known as an AV), ride service. The code simulates a scenario where passengers can leave ratings between 0.0 and 5.0 stars after their trip. The requirements say that these ratings must be stored in a Bag data structure. Doing so means order is not important, duplicates must be allowed, and the amount of reviews (ratings) left is unknown. Each time a new ride is completed, the rating is added to the Bag. The program must also be able to perform two measures for an overall review for the day. The first one is the most frequent rating (mode of the ratings) and the average rating across all rides in the day. This would simulate how AV companies would observe daily performances for their autonomous vehicles. The program must also provide a readable display of ratings, displaying how the Bag evolves as more customer reviews are added throughout the day. These requirements ensure that the program reflects a realistic AV customer review and demonstrates how a Bag data structure supports analysis of duplicate-heavy and unordered data.

2. Relevance to topic area (20 points): You must include at least one paragraph with 5 or more lines.

This program is directly relevant to the semester's project topic of Autonomous Vehicles, Bag data structures in this case. In the real world, an AV ride service would recieve large amounts of customer reviews, and the Bag data structure is well-suited for storing such data being that the amount of customer reviews are unknown. Customer ratings are a perfect example of unordered and duplicate entries where unique values are not required, making the Bag data structure a good choice to store the ratings. By using a Bag to store AV customer ratings, the program is able to take unknown entry amounts, (duplicate and overall amount of reviews), allowing to simulate realistic customer feedback reviews.

3. Implementation explanation (20 points): You must explain how you implemented the requirements with reference to the code. The references must be specific, at the level of reproducing statements or mentioning line numbers. You must include at least one paragraph with 7 or more lines.

The program is implemented through an interface and two classes. Bagm1ac is the interface for the program, the m1aclinkedlistbag is one class, and the m1actest is the testing class to test the program. In Bagm1ac, methods such as add(Item item) and size() define the essential operations for the Bag, while default implementations like isEmpty() (on line 28) and isSingleton() (on line 45) simplify usage. For my specific case I did implement them in my test

because it was unnecessary for my specific situation, but I kept them in case of future testing. The linked list bag implementation is found in m1aclinkedlistbag, where the private inner Node class stores an item and a pointer to the next node (on lines 13–21). The add method (on line 47) adds new ratings to the end of the list and increments the element count, allowing duplicate ratings to be preserved. Iteration is supported through the iterator() method (on line 65), allowing enhanced for-loops to traverse ratings. In m1actest, the testBag method (on line 12) demonstrates inserting ratings step by step, while the RatingMaxFrequency method (on line 34) calculates the most common rating, and average (on line 61) calcualting the mean rating. Finally, the main method (on line 101) ties everything together. It does so by inserting a list of sample ratings, and printing both the max frequency of the ratings, the rating that appeared the most, and average rating of the day. These implementations of the program satisfy all requirements by using a Bag data structure with some statistics that would be considered relevant to AV customer feedback.

4. Bag representation (20 points): You must explain which version of bag representation you used and why is the best for your requirements. You must include at least one paragraph with 5 or more lines.

For this assignment, a linked list Bag was chosen because it best fits the AV customer reviews scenario. Since the number of ratings is unknown for the day, the linked list bag data structure can grow without bound. While a fixed bag data structure would likely overflow resulting in an error. The linked list bag design allows for dynamic memory, allowing ratings to be added without resizing the bag capacity. Adding reviews is efficient because each new rating is added in constant time by updating the endNode. Iteration with the linked list Bag is easy with the support from nested loops calculating the mode and average for reviews. Overall, the linked list Bag efficiently handles customer ratings, especially in the context of an autonomous vehicle customer feeback.

Execution (20 points)

```
Creative Assignment - M1A-C - by Matthew Novak
Executed on: Thu Sep 04 21:21:19 EDT 2025
Customer Ratings for the day from 0.0 stars to 5.0 stars
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Ratings for the day: []
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 3.5 stars
Ratings for the day: [3.5]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 5.0 stars
Ratings for the day: [3.5 stars, 5.0]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 4.3 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 1.1 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 4.5 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 3.5 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 2.5 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5 stars, 2.5]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 5.0 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5 stars, 2.5 stars, 5.0]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 3.5 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5 stars, 2.5 stars, 5.0 stars, 3.5]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
A customer gave a rating of: 4.5 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5 stars, 2.5 stars, 5.0 stars, 3.5 stars, 4.5]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

A customer gave a rating of: 2.0 stars
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5 stars, 2.5 stars, 5.0 stars, 3.5 stars, 4.5 stars, 2.0]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Ratings for the day: [3.5 stars, 5.0 stars, 4.3 stars, 1.1 stars, 4.5 stars, 3.5 stars, 2.5 stars, 5.0 stars, 3.5 stars, 4.5 stars, 2.0]
The rating you recieved the most today was: 3.5 stars. You recieved that rating 3 times today.
Your average rating for the day was: 3.6 stars.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```