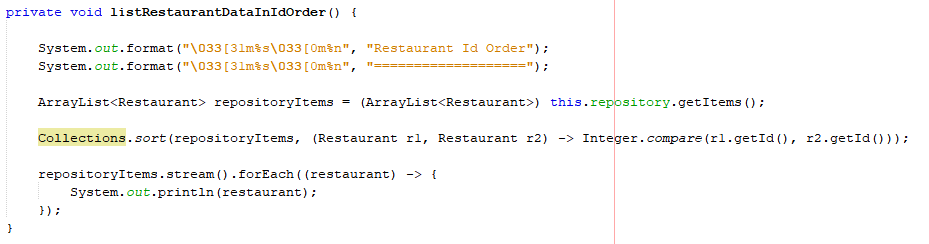
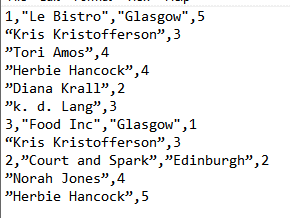
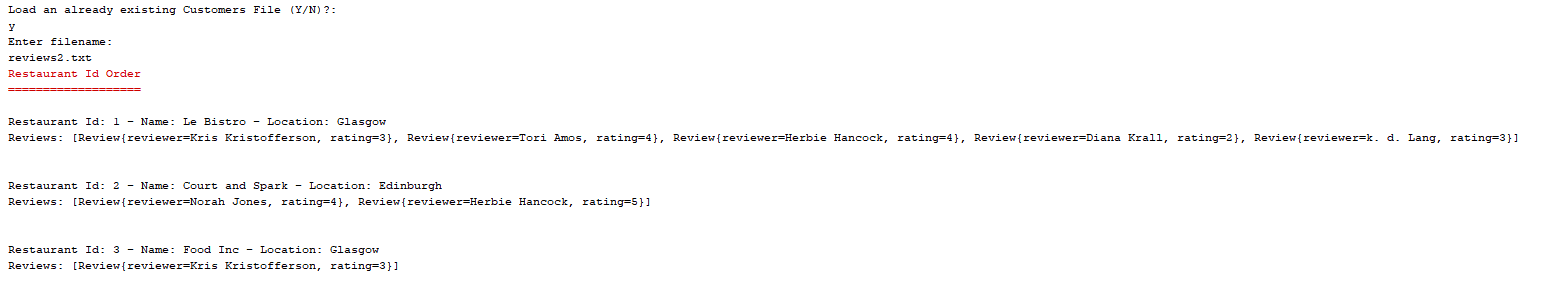
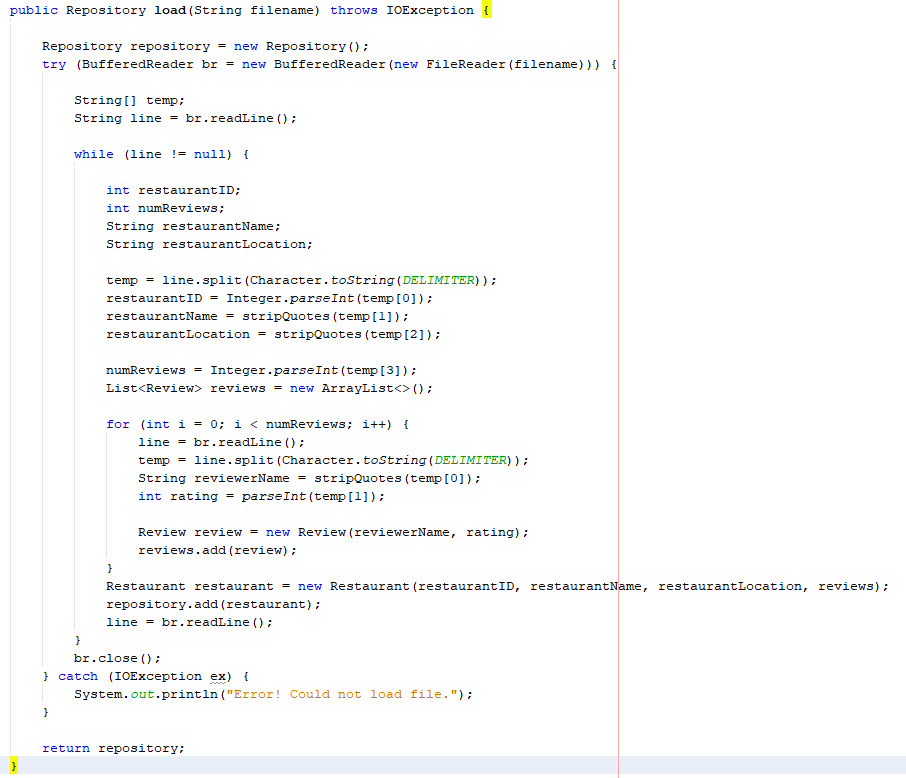
**Increment 1**







This increment functions by firstly loading in the text file supplied by the user, and it loops through until the end of the text file. It splits the first line by the comma and saves each split area to a string array and assigns each position to a variable name according to the type of data it is. Position 0 would be the ID, 1 being name, 2 being location, and the number of reviews being position 3.

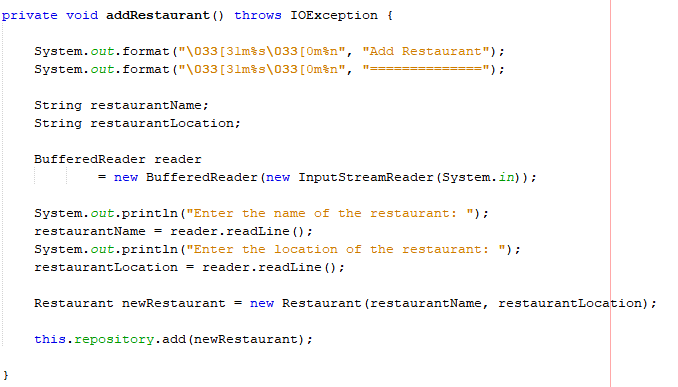
A new List is created of object Review called reviews, which is followed by a for loop to loop through the next set amount of lines (the amount is determined by the number of views variable we got before) and then the line is read in and split by the comma and stored in the String array, and each split part is stored to a variable. Position 0 being the reviewer name and position 1 being the rating. These two variables are used to create a new review, and this is added onto the previously created List of reviews.

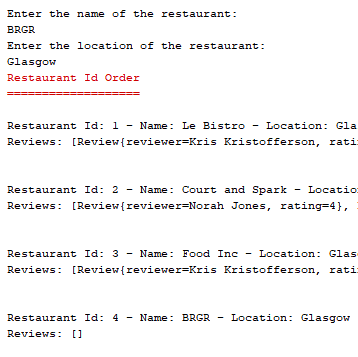
The previous variables that we set of the id, name, location, and now the list of reviews, are all used to create a Restaurant object called restaurant, and this newly created object is added onto a Repository object. This loops through until all of the restaurants have been read in (until end of file) and then it returns the repository object.

This repository object calls getItems() and stores this to an ArrayList of object Restaurant.

Collections takes two items from repositoryItems, and it runs the compare method against each other recursively and then re-orders in ascending order. Then for each restaurant in repositoryItems is printed out.

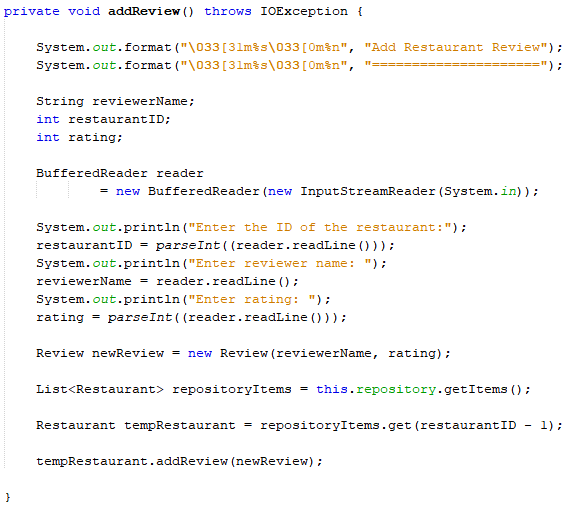
**Increment 2**

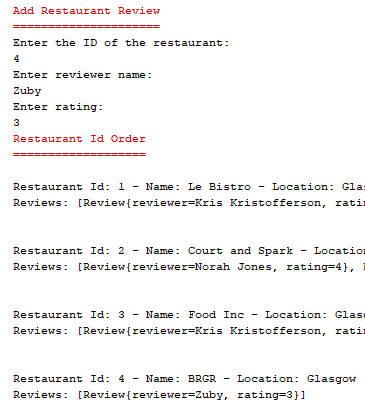




This method takes in the user inputs and stores the first input to a String called restaurantName and the second input to a String called restaurantLocation. This is then used to create a new Restaurant object by passing in those two variables, and then this new Restaurant is added onto the end of the Repository object.

**Increment 3**

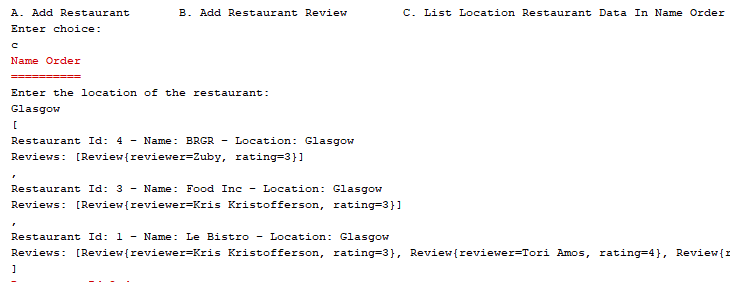
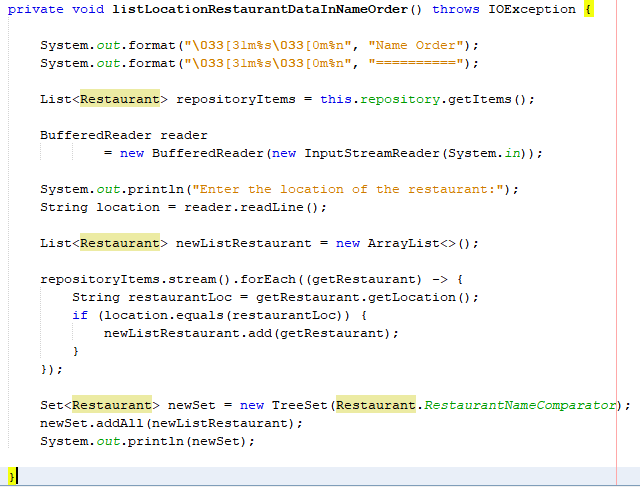




This method takes reads in three lines and stores them to variables. Line 1, being parsed in as an integer, is stored as the restaurantID, line 2 as a String named reviewerName, and line 3 which is parsed as an integer which is the rating.

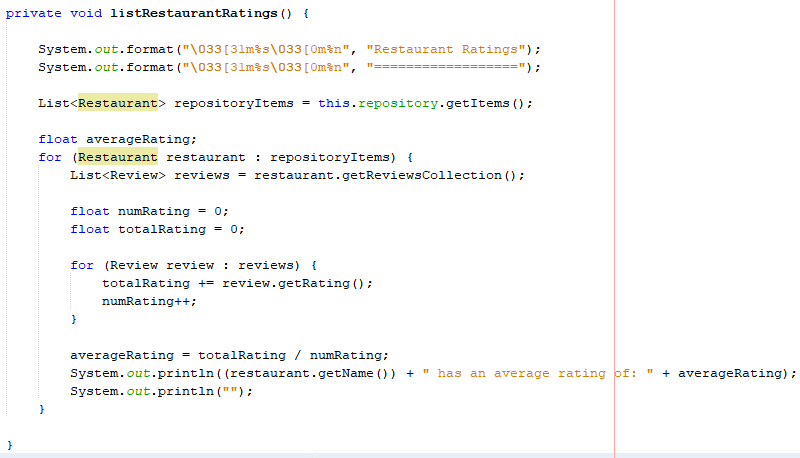
The reviewerName and rating is used to create a new Review object and the restaurantID is used to find the position of the restaurant in repositoryItems and adds that restaurant to a new Restaurant object called tempRestaurant, and the review is finally added onto that tempRestaurant.

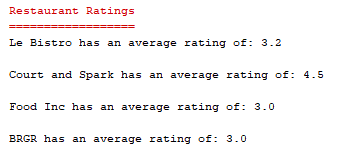
**Increment 4**



This method stores the current items in the Repository object into a List of object type Restaurant and takes in a String input which is stored to the variable ‘location’. Then then repositoryItems is streamed one object at a time, and for each of those objects you get the current object which is shown as ‘getRestaurant’ and then for that current restaurant you get the variable Location stored under it, and then you check if that variable Location is the exact same as the location the user input. If so then that current restaurant is then added to a new List of type Restaurant called newListRestaurant. A new Set of type Restaurant is created and then the newListRestaurant is added to that set, which does the RestaurantNameComparator and orders the objects in that list in alphabetical order.

**Increment 5**





This method looks at each restaurant in repositoryItems, and for each restaurant it calls getReviewsCollection and stores that to a List of object type Review, and then for each review in that list of reviews it calls getRating which returns an int value, which is added onto totalRating, and the numRating is incremented by 1 to keep track of the number of reviews in that list.

Then outside of that nested loop the average is calculated, and then the restaurant name is printed out with the average rating.

**Increment 6**

