ACS-2947-050
Assignment #4
Due by Saturday December 1 at 11:59 pm

- Submit your .java files to 2947L-070@acs.uwinnipeg.ca or 2947L-071@acs.uwinnipeg.ca
- Include your name and student number in each file as a comment
    - Document the Orders class using Javadoc notation
    - Include comments as needed
    - Use appropriate exception handling where necessary

Part A - Linear Probing

Develop a program named HighestWordFrequency that determines the 10 most frequently used words found in PartA.txt[1]. Note that your program must exclude all sentence punctuation characters when parsing.

1. Using the Map and Entry interfaces and AbstractMap class from Lab 7, provide the ProbeHashMap implementation. Use the AbstractHashMap class from your notes/text as a base.
    - Use your ProbeHashMap to store each word and its count
2. Create a class named MergeSort that uses the merge-sort algorithm and a non-generic EntryComparator to sort items in *descending* order
    - Sort your list of word frequencies, and display the top 10 result

PART B - Separate Chaining

Create a version of the ChainHashMap that uses Java's ArrayList for each bucket.

1. Implement the Map interface using the interfaces and abstract classes from Part A. Name your class ALChainHashMap.
    - Use an ArrayList as the auxiliary data structure that holds entries of colliding keys
    - Add a method named numCollisions that returns an integer representing the number of collisions that occurred in your hashmap.

[1] Rowling, J.K. Harry Potter & the Sorcerer's Stone . Scholastic, 1998
http://harrypotter.scholastic.com/excerpts/HP_Book1_Chapter_Excerpt.pdf

2. Create a class named `Order` that stores:
    - strings `orderID` and `customerID`
    - double `amount`

3. In a driver class called `PartB`, create a hash map of Orders.
    - read in the [`PartB.txt`]() file
        - set each line as an instance of `Order`
        - create a `mapEntry` of each order instance, using `orderID` as `key` and `Order` instance as `value`

4. In your output, display
    - The list of orders
    - The number of orders
    - The number of collisions that occured in the hashmap