Maps

Implement a method which, given a list of transactions at a certain kind of establishment on Sixth Street, returns the total tabs for each customer. The transactions will be passed into this method as a List<String> where each string will have the following format: <Name> <Cost> (Name will be a single word, Cost will be a double). Your method will return a Map which maps customers names to their total tab.

Complete the following method.

// Calculates the total tabs for each customer given a list of transactions
// pre: transactions != null
// post: A map which maps customer names to their total tab.
public static Map<String, Double> sumTabs(List<String> transactions) {

Here are some example calls to sumTabs(). (The resulting map does not have to be sorted like shown):

transactionList1 = ["A 5.50", "B 7.75", "A 7.90", "A 15.30", "B 2.25"]
sumTabs(transactionList1).toString() → {A=10.0, B=28.7}

transactionList2 = ["A 1.25", "B 10.15", "C 8.25", "D 7.00"]
sumTabs(transactionList1).toString() → {A=1.25, B=10.15, C=8.25, D=7.0}

You may create a TreeMap or HashMap. You may also use the Java Scanner class’s constructor, next(), and nextDouble() methods, or any String methods and the Double class’s parseDouble() method.

Do not use any other Java classes or methods.
// Calculates the total tabs for each customer given a list of transactions
// pre: transactions != null
// post: A map which maps customer names to their total tab.
public static Map<String, Double> sumTabs(List<String> transactions) {