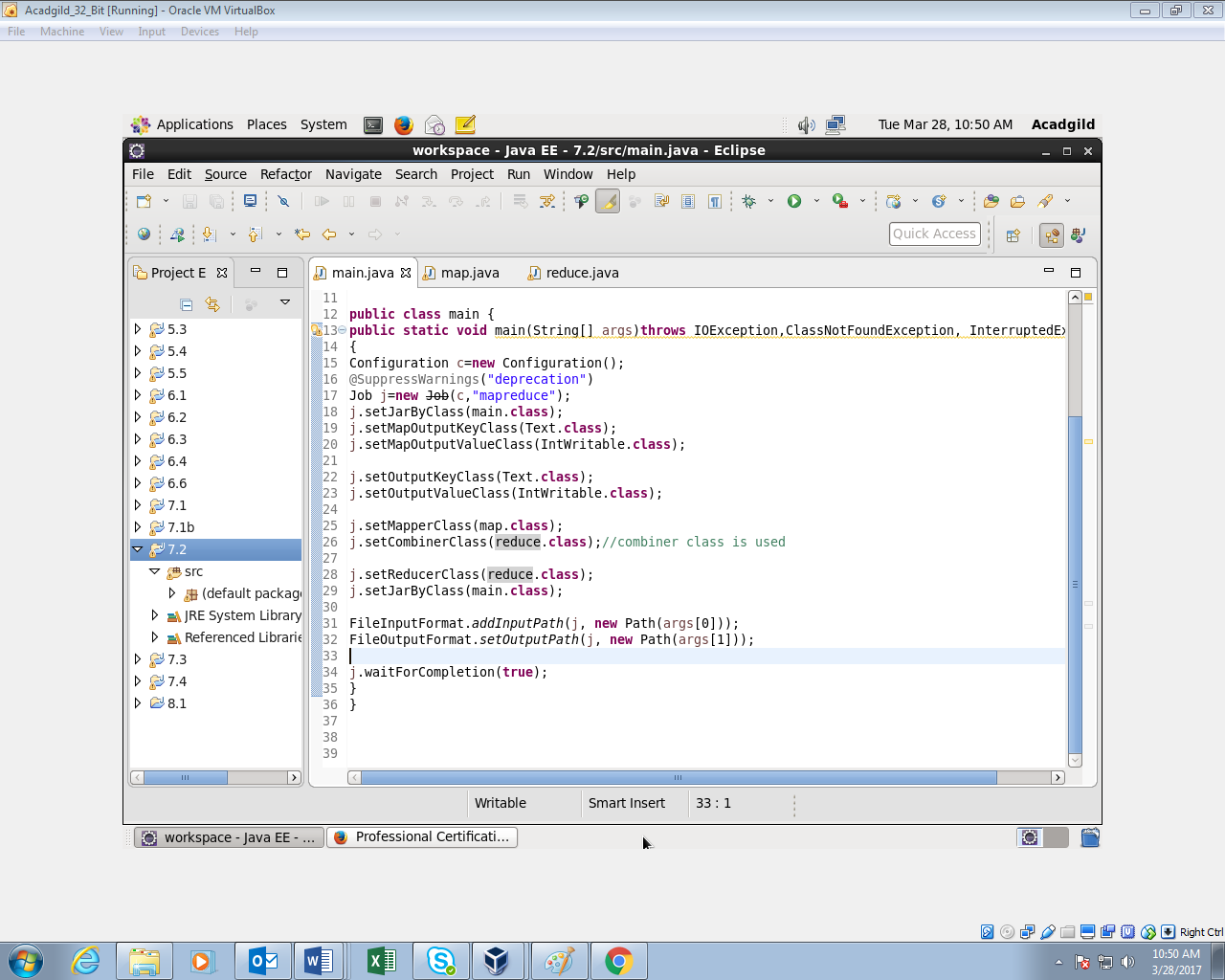
Sales of different TV Task 6 : Modify Sales of different TV Task 3 (refer session 5, assignment 2) to take advantage of Combiner.

Sales of different TV Task 7 : Write a Mapreduce program to view the total sales for each product for every Company corresponding to each size. Make sure that all records for a single company goes to a single reducer and inside every reducer, keys must be sorted in descending order of the size. You may write a custom WritableComparable for this purpose. "

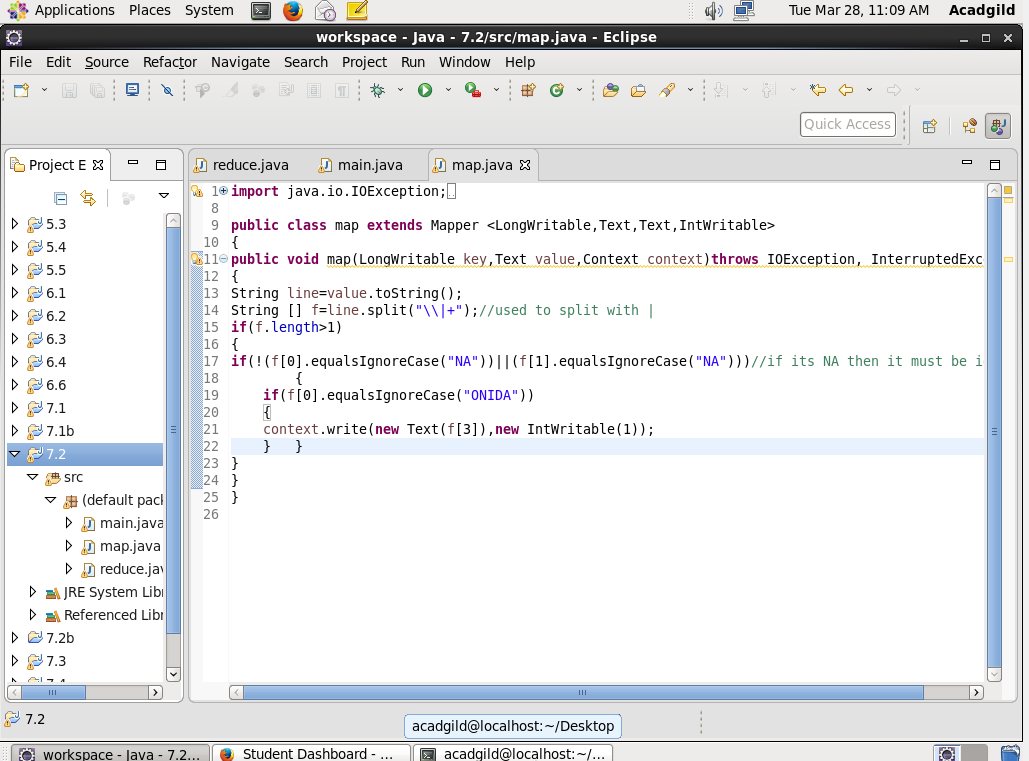
DRIVER: Using combiner and reducer class since both have the same logic.

Main



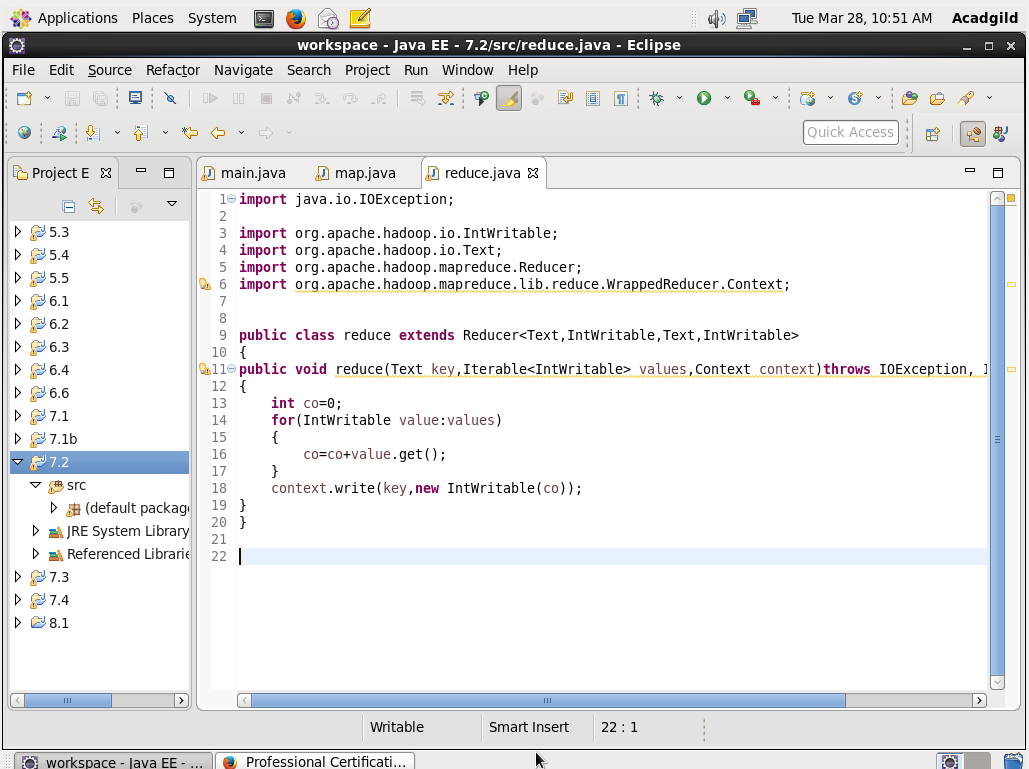
Map

Will remove all the records which contain “NA” and later will filter the records with ONIDA will send the key as company name for each key.

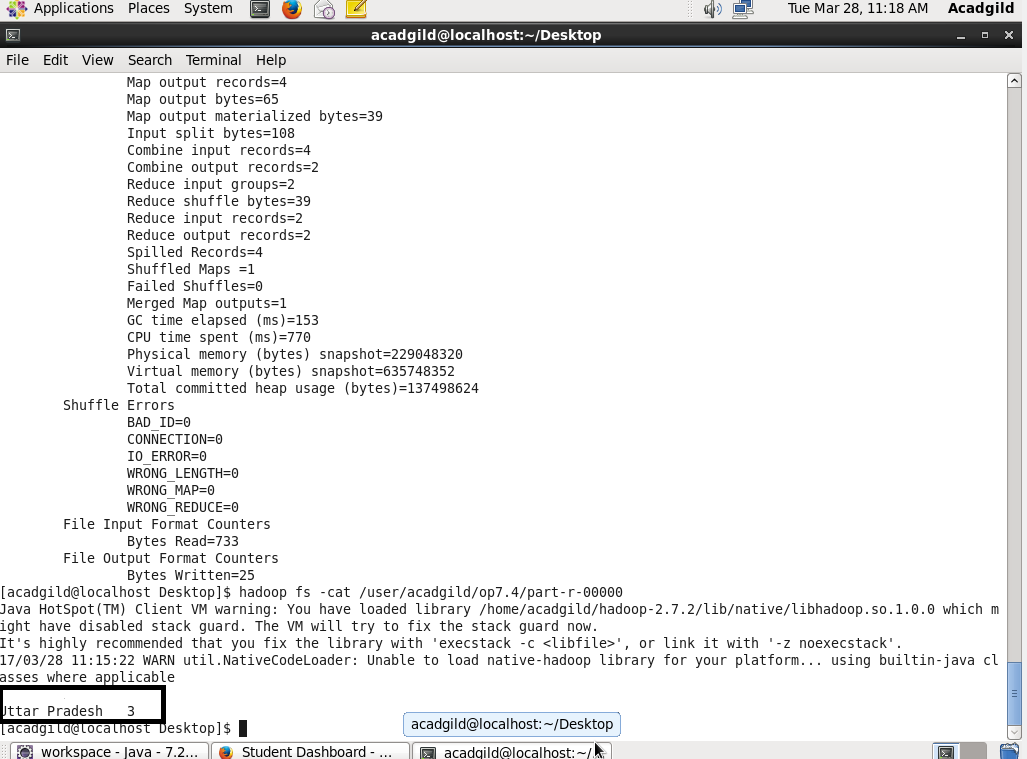


Reduce

Will count the number of onida.

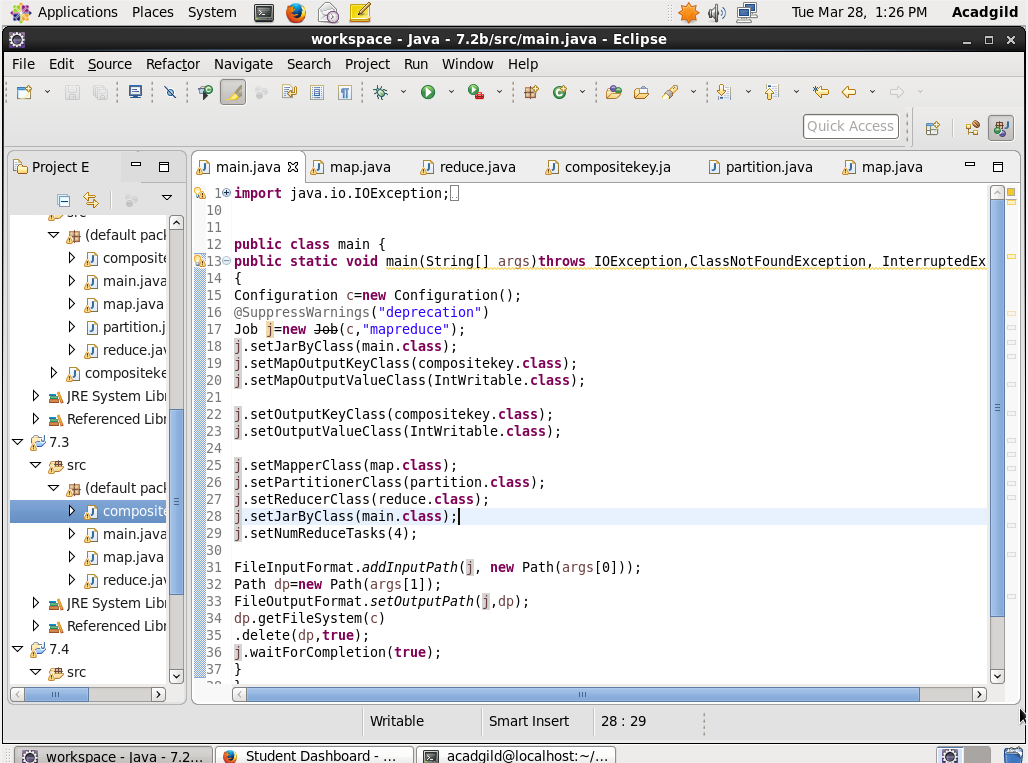


Output



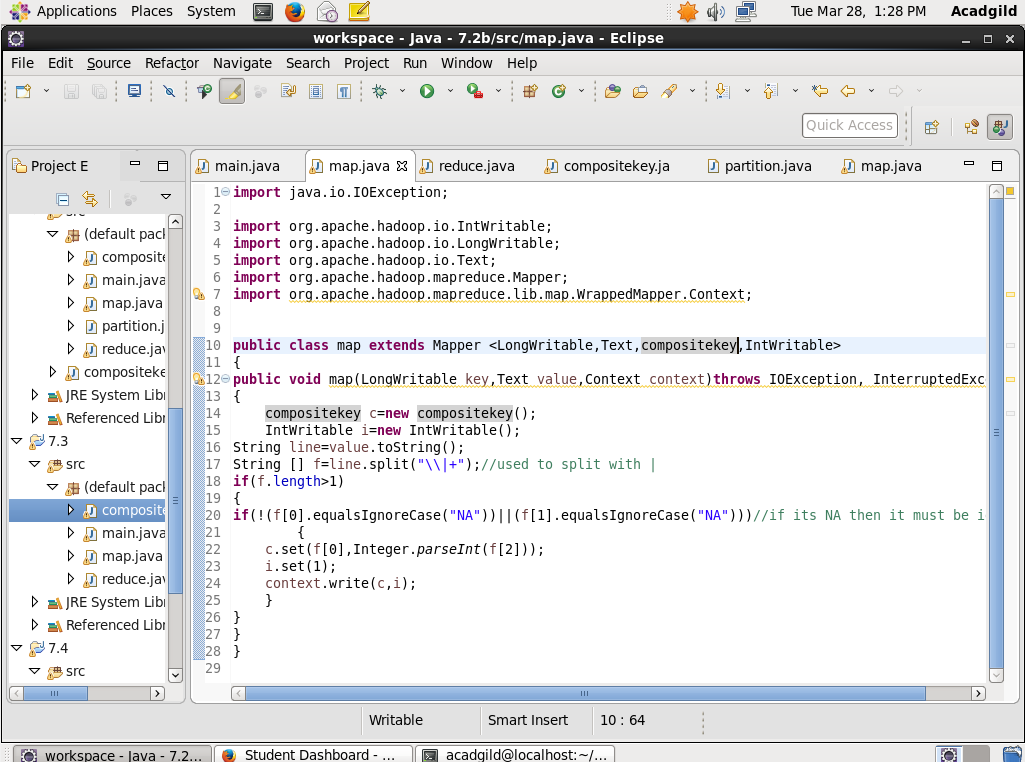
QUESTION 2:

Main



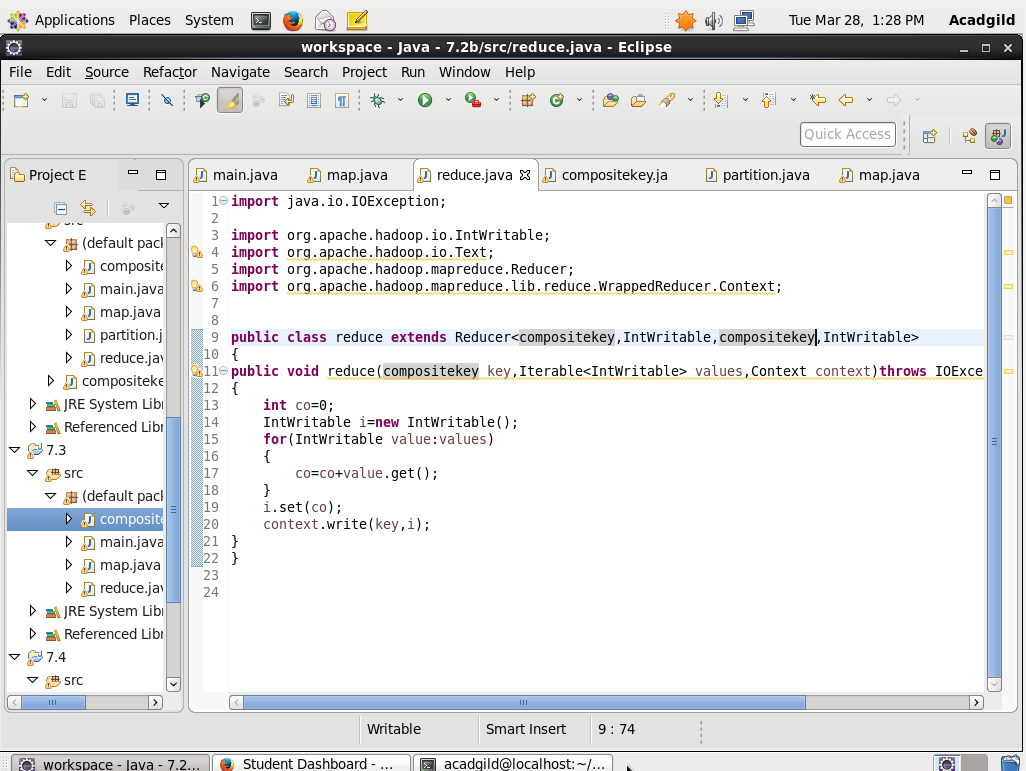
Map

This will split the data by | and will send the size and company and will count 1



Reduce

Will iterate and count the values.



Composite key

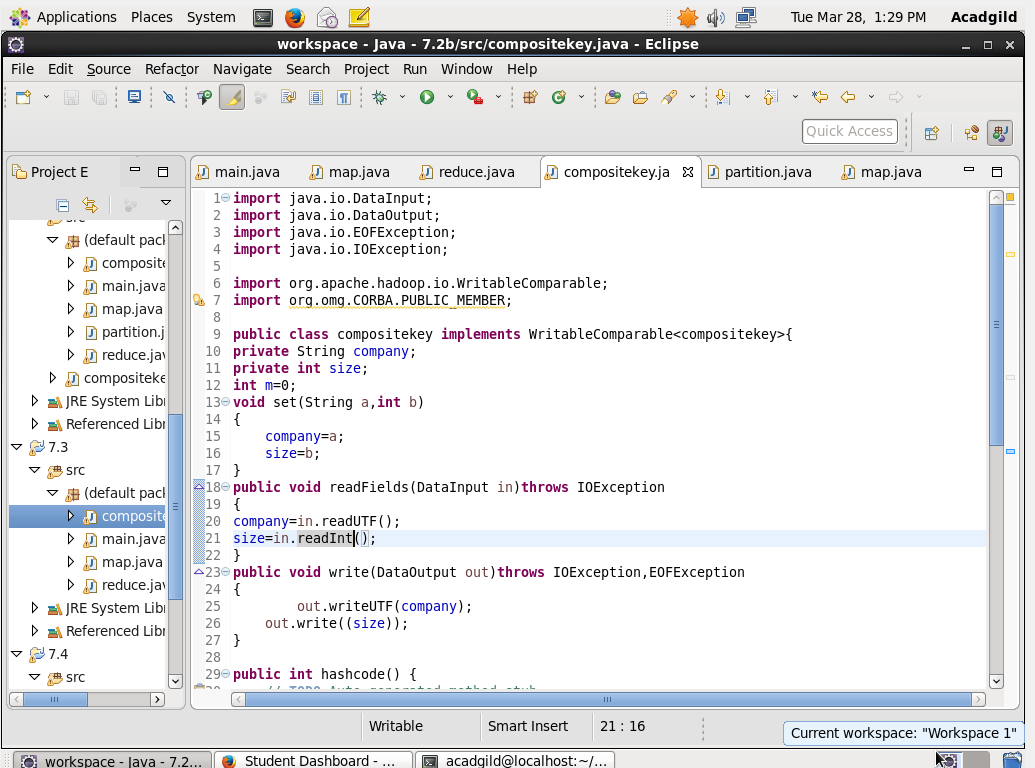
Step 1:while overriding will return the hash code of the company

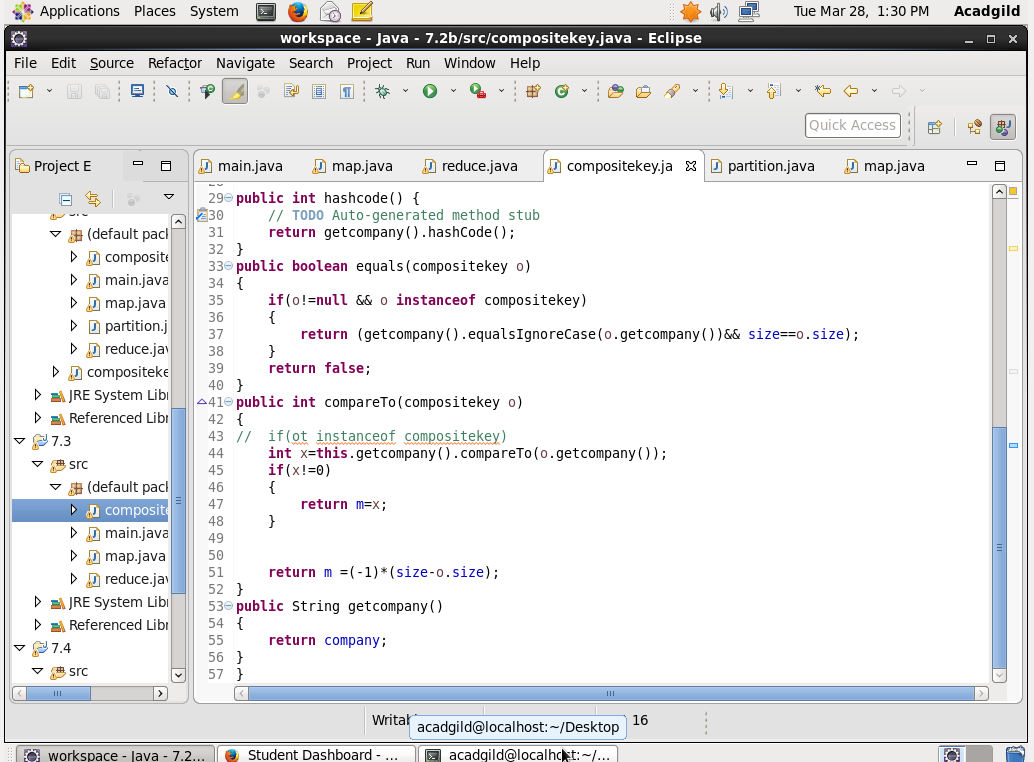
Step 2: OVERRIDING EQUALS METHOD: will check whether the company and size are equal. If it founds to be equal then will send the location of the hash code or else it will compare again.

Step 3 : OVERRIDING COMPARE TO METHOD: if both the company name are equal then it will return -1.(returning -1 will help to return in descending order).

Step 4 : OVERRIDING TOSTRING METHOD :it is to print the size and the company

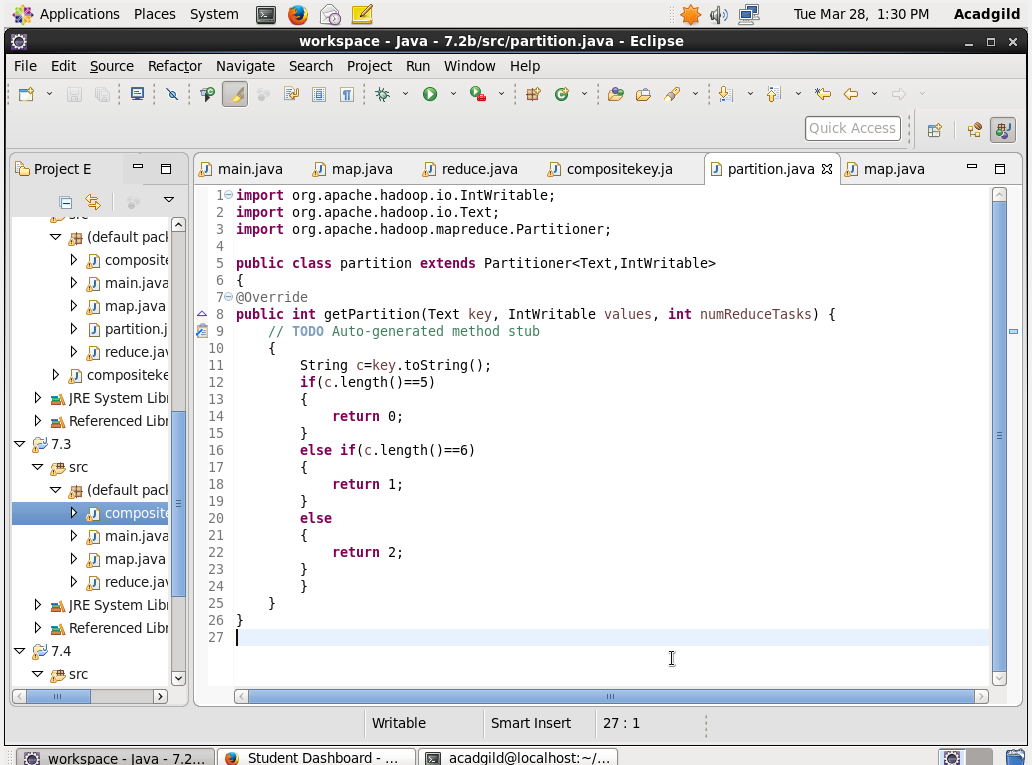
Readfields and writefields are used for serealisation and deserealisation



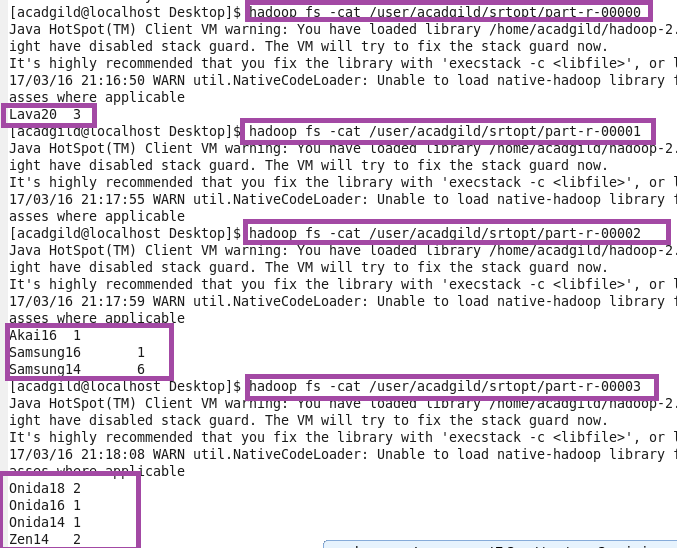


Partition key

In parttitioner depending on company name hashcode divide company hachcode %numreduce task so that same company records go to the same reducer



Output

****