Assignment 9

```
package stream;

import java.util.Arrays;

import java.util.Comparator;

import java.util.List;

import java.util.Map;

import java.util.stream.Collectors;

class Fruits{

	String name;

	int calories;

	int price;

	String color;

	public Fruits(String name, int calories, int price, String color) {

		super();

		this.name = name;

		this.calories = calories;

		this.price = price;

		this.color = color;

	}

	@Override

	public String toString() {

		return "Fruit [name=" + name + ", calories=" + calories + ", price=" + price + ",
color=" + color + "]";

	}

}

public class StreamDemo {


	public static void main(String[] args) {


		List<Fruits> fruitList = Arrays.asList(

				new Fruits("Apple", 150 , 10, "Red"),
```

```
                    new Fruits("Orange", 60 , 30, "Blue"),
                    new Fruits("banana", 30 , 20, "Red"),
                    new Fruits("Watermelon", 180 , 50, "Blue")
                    );

List<News> newsList = Arrays.asList(
                    new News(1, "E" , "I", "well"),
                    new News(2, "F" , "J", "good"),
                    new News(1, "F" , "K", "great"),
                    new News(4, "H" , "I", "excellent")

                    );

List<Trader> traderList = Arrays.asList(
                    new Trader("O", "madurai"),
                    new Trader("N", "chennai"),
                    new Trader("M", "tiruchy"),
                    new Trader("P", "pudhukottai"),
                    new Trader("Q", "dindigul")
                    );

List<Transaction> transactionList = Arrays.asList(
                    new Transaction(traderList.get(0), 2000, 1000),
                    new Transaction(traderList.get(1), 2011, 8000),
                    new Transaction(traderList.get(2), 2011, 3000),
                    new Transaction(traderList.get(3), 2003, 6000)
                    );



// 1st Ques
System.out.println("Stream First Question output");
```

```
fruitList.stream().filter(l -> l.calories<100).forEach(l -> System.out.println(l.name));



// 2nd Ques

System.out.println("\n"+"Stream Second Question output");

fruitList.stream().sorted(Comparator.comparing(l -> l.color)).forEach( l->
System.out.println(l));



// 3rd Ques

System.out.println("\n"+"Stream 3rd Question output");

fruitList.stream().filter(l ->
l.color.equalsIgnoreCase("Red")).sorted(Comparator.comparingInt(l-> l.price))

.forEach(System.out::println);



// 4th Ques

System.out.println("\n"+"Stream 4th Question output");

newsList.stream().collect(Collectors.groupingBy(l -> l.newsId, Collectors.counting()))

.entrySet().stream().max(Map.Entry.comparingByValue())

.ifPresent(l-> System.out.println("News Id : "+ l.getKey() + " has the maxium comment i.e.
:" + l.getValue()));



// 5th Ques

System.out.println("\n"+"Stream 5th Question output");

newsList.stream().filter(l->
l.comment.equalsIgnoreCase("Budget")).collect(Collectors.groupingBy(l -> l.comment,
Collectors.counting()))

.entrySet().stream().max(Map.Entry.comparingByValue())

.ifPresent(l-> System.out.println( l.getKey() + " are arrived " + l.getValue() + " times"));
```

```
		// 6th Ques
		System.out.println("\n"+"Stream 6th Question output");
		newsList.stream().collect(Collectors.groupingBy(l->l.commentByUser, Collectors.counting()))
		.entrySet().stream().max(Map.Entry.comparingByValue())
	.ifPresent(l-> System.out.println("User Id : "+ l.getKey() + " has did the maximum comment i.e. :" + l.getValue()));


		// 7th Ques
		System.out.println("\n"+"Stream 7th Question output");
		newsList.stream().collect(Collectors.groupingBy(l->l.commentByUser, Collectors.counting()))
		.entrySet().stream()
		.forEach(l -> System.out.println(l));


		// 8th Ques
		System.out.println("\n"+"Stream 8th Question output");
		transactionList.stream().filter(l -> l.year == 2011).sorted(Comparator.comparingInt(l-> l.value))
		.forEach(l -> System.out.println(l));


		// 9th Ques
		System.out.println("\n"+"Stream 9th Question output");
		traderList.stream().map(l-> l.city.toLowerCase()).distinct().forEach(l -> System.out.println(l));


		// 10th Ques
		System.out.println("\n"+"Stream 10th Question output");
```

```
		traderList.stream().filter(l -> l.city.equalsIgnoreCase("Pune")).sorted(Comparator.comparing(l -> l.name))
		.forEach(l -> System.out.println(l));



		// 11th Ques
		System.out.println("\n"+"Stream 11th Question output");
		traderList.stream().sorted(Comparator.comparing(l -> l.name)).map(l -> l.name).forEach(System.out::println);



		// 12th Ques
		System.out.println("\n"+"Stream 12th Question output");
		traderList.stream().filter(l -> l.city.equalsIgnoreCase("Indore")).forEach(System.out::println);



		// 13th Ques
		System.out.println("\n"+"Stream 13th Question output");
		transactionList.stream().filter(l-> l.trader.city.equalsIgnoreCase("Delhi")).forEach(System.out::println);



		// 14th Ques
		System.out.println("\n"+"Stream 14th Question output");
		transactionList.stream().max(Comparator.comparingInt(l-> l.value)).ifPresent(System.out::println);

		// 15th Ques
		System.out.println("\n"+"Stream 15th Question output");
		transactionList.stream().min(Comparator.comparingInt(l-> l.value)).ifPresent(System.out::println);
```

```
	}
}
```

```
class News{
	int newsId;
	String postedByUser;
	String commentByUser;
	String comment;
	public News(int newsId, String postedByUser, String commentByUser, String comment) {
		super();
		this.newsId = newsId;
		this.postedByUser = postedByUser;
		this.commentByUser = commentByUser;
		this.comment = comment;
	}
}

class Trader{
	String name;
	String city;
	public Trader(String name, String city) {
		super();
		this.name = name;
		this.city = city;
	}

	@Override
```

```
	public String toString() {


		return name+" "+ city;

	}
}


class Transaction{
	Trader trader;
	int year;
	int value;
	public Transaction(Trader trader, int year, int value) {
		super();
		this.trader = trader;
		this.year = year;
		this.value = value;
	}
	@Override
	public String toString() {
		 return trader +" "+year+ " " +value ;
	}
}
```

Output

```
Stream First Question output
Orange
banana

Stream Second Question output
Fruit [name=Orange, calories=60, price=30, color=Blue]
Fruit [name=Watermelon, calories=180, price=50, color=Blue]
Fruit [name=Apple, calories=150, price=10, color=Red]
Fruit [name=banana, calories=30, price=20, color=Red]

Stream 3rd Question output
Fruit [name=Apple, calories=150, price=10, color=Red]
Fruit [name=banana, calories=30, price=20, color=Red]

Stream 4th Question output
News Id : 1 has the maxium comment i.e. :2

Stream 5th Question output

Stream 6th Question output
User Id : I has did the maximum comment i.e. :2

Stream 7th Question output
I=2
J=1
K=1

Stream 8th Question output
M tiruchy 2011 3000
N chennai 2011 8000

Stream 9th Question output
madurai
chennai
tiruchy
pudhukottai
dindigul

Stream 10th Question output

Stream 11th Question output
M
```