STREAM API – JAVA 8

(1) From list print only duplicate numbers?

```
public class DupLicateNumPrint {
public static void main(String[] args) {

List<Integer> duplicateNum = Arrays.asList(12,43,12,53,64,88,98,98);
List<Integer> duplicateNumbers = duplicateNum.stream()
.filter(i -> duplicateNum.stream().filter(j -> j.equals(i)).count() > 1)
.distinct()
.collect(Collectors.toList());
System.out.println("Duplicate numbers: " + duplicateNumbers);
}
```

(2) sort by employee name whose salary greater than 15000 and print their name in upperCase with 10% hike?

```
package steam_api_filter;
public class Employee {
private String name;
private String city;
private double salary;
public Employee(String name, String city, double salary) {
super();
```

```
this.name = name;
this.city = city;
this.salary = salary;
}
@Override
public String toString() {
return "Employee [name=" + name + ", city=" + city + ", salary=" + salary +
"]";
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public String getCity() {
return city;
}
public void setCity(String city) {
this.city = city;
}
public double getSalary() {
return salary;
}
public void setSalary(double salary) {
```

```
this.salary = salary;
}
}


package steam_api_filter;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

import steam_api_map.Eployee;

public class EmployeeImpl {
public static void main(String[] args) {


        Employee e1 = new Employee("ratan","banglore",85000);
        Employee e2 = new Employee("Ramesh","delhi",5000);
        Employee e3 = new Employee("punit","mumbai",14000);
        Employee e4 = new Employee("sam","patna",4000);
        Employee e5 = new Employee("Arman","kolkata",95000);

        List<Employee> employees = Arrays.asList(e1,e2,e3,e4,e5);
```

```
        employees.stream()
        .sorted((name1, name2) ->
name1.getName().compareTo(name2.getName()))
        .filter(x -> x.getSalary() > 15000)
        .map(n -> {
                n.setSalary(n.getSalary() * 1.1); // Multiply salary by 1.1 (10%
increase)
                return n;
        })
        .forEach(e -> System.out.println(e.getName().toUpperCase() + "=" +
e.getSalary()));




}



}
```

(3) Print even numbers?

```
package steam_api_filter;
```

```
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class EvenNumbers {
public static void main(String[] args) {
        List<Integer> evenNumbers = Arrays.asList(1,2,5,6,9,8,4,7,11,10);

evenNumbers.stream().filter(val->val%2==0).collect(Collectors.toList()).fo
rEach(System.out::println);;

}
}
```

(4) Print odd numbers?

```
package steam_api_filter;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class OddNumbers {
public static void main(String[] args) {
```

```
        List<Integer> OddNumbers = Arrays.asList(5,6,3,101,20,9,14,8,7);
        OddNumbers.stream().filter(val->val%2!=0).collect(Collectors.toList
()).forEach(System.out::println);;
}
}
```

(5) from a string print only those name which start with s and p in sorting order?

```
package steam_api_filter;

import java.util.Arrays;
import java.util.List;

public class NameFiltringwithSortingWithDistinct {
public static void main(String[] args) {



        List<String> names =
Arrays.asList("sahil","punit","punit","satish","satish","pulkit","arman","r
amesh");
       names.stream().filter(name->name.startsWith("s") ||
name.startsWith("p")).sorted().distinct().forEach(System.out::println);
}
}
```

(6) print only even number till 8 out of 20?

```
package steam_api_filter;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class OnlyEvenNumTillCond {
public static void main(String[] args) {
        List<Integer> list = Arrays.asList(2,4,3,7,9,10,6,8);
        list.stream().filter(val->val%2==0 &&
val<=8).collect(Collectors.toList()).forEach(System.out::println);;
}
}
```

(7) sort by city and find the patient name,gender,dieses whose bill is greater then 5000 ….many more ??

```
package steam_api_filter;
public class Patient{
private long id;
private String name;
private String city;
private String dieses;
```

```
private String gender;
private int bill;
public Patient(long id, String name, String city, String dieses, String
gender, int bill) {
super();
this.id = id;
this.name = name;
this.city = city;
this.dieses = dieses;
this.gender = gender;
this.bill = bill;
}
public long getId() {
return id;
}
public void setId(long id) {
this.id = id;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public String getCity() {
```

```
return city;
}
public void setCity(String city) {
this.city = city;
}
public String getDieses() {
return dieses;
}
public void setDieses(String dieses) {
this.dieses = dieses;
}
public String getGender() {
return gender;
}
public void setGender(String gender) {
this.gender = gender;
}
public int getBill() {
return bill;
}
public void setBill(int bill) {
this.bill = bill;
}
@Override
```

```
public String toString() {
return "A [id=" + id + ", name=" + name + ", city=" + city + ", dieses=" + dieses + ", gender=" + gender
+ ", bill=" + bill + "]";
}
}
```

```
package steam_api_filter;

import java.util.Arrays;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

public class PatientImpl {

    public static void main(String[] args) {
        Patient a = new Patient(101,"rana","banglore","corona","male",1000);
        Patient a1 = new Patient(102,"rohit","mumbai","fever","female",13000);
```

```
        Patient a2 = new
Patient(103,"sam","delhi","heart_attack","female",2000);

        Patient a3 = new
Patient(104,"nehal","banglore","stone","male",4000);

        Patient a4 = new
Patient(105,"amit","kolkata","apendice","female",6000);

        Patient a5 = new
Patient(106,"karan","delhi","lever","male",7000);

        Patient a6 = new
Patient(107,"ritesh","pune","corona","female",7000);



        //sort by city and find the patient name,gender,dieses whose
bill is greater then 5000


        List<Patient> list = Arrays.asList(a,a1,a2,a3,a4,a5,a6);


    list.stream().sorted(Comparator.comparing(Patient::getCity)).filter(
m->m.getBill()>5000)


    .forEach(e->System.out.println(e.getName()+"="+e.getBill()+"="+e.
getGender()+"="+e.getDieses()));;



        // give how many males are in the list


        long    maleGender=
list.stream().filter(male->male.getGender().equals("male")).count();
```

```
                System.out.println("male gender:"+maleGender);



                // print how many females are in the list
                long femaleGender =
list.stream().filter(female->female.getGender().equals("female")).count();
                System.out.println("female gender:"+femaleGender);



                //group the city in sorting order
                Map<String, List<Patient>> collect =
list.stream().sorted(Comparator.comparing(Patient::getCity)).collect(Colle
ctors.groupingBy(Patient::getCity));
                                        System.out.println(collect);
        }

}
```

(8) take a list of names and print those names which ends with letter e?

```
package steam_api_filter;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;
```

```
public class PrintNameEndWithE {


	public static void main(String[] args) {


		// take a list of names and print those names which ends with letter e


		List<String> names = Arrays.asList("apple","english","rate","orange","mango");

		List<String> collect = names.stream().filter(name->name.endsWith("e")).collect(Collectors.toList());

		System.out.println(collect);

	}


}
```

(9) write a program using equalIgnoreCase?

```
package steam_api_filter;


import java.util.Arrays;

import java.util.List;

import java.util.stream.Collectors;
```

```
public class UsageOfEqualsIgnoreCase {
public static void main(String[] args) {
        List<String> names =
Arrays.asList("meraz","mohit","Meraz","diler");



        //output:    MERAZ, MERAZ
        names.stream().filter(name->name.equalsIgnoreCase("meraz")).ma
p(String::toUpperCase).collect(Collectors.toList()).forEach(System.out::pri
ntln);;


}
}
```

(10) write a program using flatmap?

```
package steam_api_Flatmap;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class FlatMapp {
public static void main(String[] args) {
        List<String> names = Arrays.asList("hello","welcome");
```

```
        List<String> flatMap =
names.stream().flatMap(name->Arrays.stream(name.split(""))).map(Strin
g::toUpperCase).collect(Collectors.toList());

                System.out.println(flatMap);

}

}
```

(11) write a program for usage of map?

```
package steam_api_map;


import java.util.Arrays;

import java.util.List;


public class UsageOfMapPrintingNumbers {

public static void main(String[] args) {

        List<Integer> numbers = Arrays.asList(1,2,5,6,8,4);//1*1=1,
2*2=4,5*5=25.....146


        // multiplying and then collecting sum

        Integer sum =
numbers.stream().map(number->number*number).reduce(0,
Integer::sum);

        System.out.println(sum);

}

}
```

(12) print ascue number of string?

```
package utility_package;

import java.util.List;
import java.util.stream.Collectors;

public class AscueNumber {
public static void main(String[] args) {
        //how to convert in ascue number
        String str="sahil";

        //for integer value
        str.chars().forEach(System.out::println);;

        //for character
        List<Character> collect =
str.chars().mapToObj(name->(char)name).collect(Collectors.toList());
        System.out.println(collect);



}
}
```