```
19-may-2022        STREAM API-JAVA8
-----------        ----------------
```

```
package Java8Features;
import java.util.ArrayList;
class Product
{
    int id;
    String name;
    double price;
    public Product(int id, String name, double price) {
        super();
        this.id = id;
        this.name = name;
        this.price = price;
    }

}
public class StreamTestProduct {

    public static void main(String[] args) {
    ArrayList<Product> al=new ArrayList<Product>();
    al.add(new Product(1,"Dell Laptop",45000.2));
    al.add(new Product(2,"HP Laptop",55000.2));
    al.add(new Product(3,"LENOVA Laptop",25000.2));
    al.add(new Product(4,"APPLE Laptop",90000.2));
    al.stream()
    .filter(p->p.id>=2)
    .forEach(p->System.out.println(p.name));


    }

}
```

```
OUTPUT:
-------
HP Laptop
LENOVA Laptop
APPLE Laptop
------------------------------
```

2.THE TASK IS TO COLLECT THE NEGATIVE EVEN NUMBERS FROM THE GIVEN ARRAYLIST AND STORE THE VALUES INTO NEW ARRAY;LIST BY USING FILTER AND COLLECTOR METHOD.

```
package Java8Features;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class StreamTest1 {

    public static void main(String[] args) {
        List<Integer> al=Arrays.asList(1,-2,3,4,-6,-5,-4);
        List<Integer>nl=al.stream()
        .filter(p->((p<0)&&(p%2==0)))
        .collect(Collectors.toList());
         System.out.println(nl);
    }

}
```

OUTPUT:

```
[-2, -6, -4]
```

3.GET AN EMPLOYEE DEATAILS AND SET THE LOCATION AS PUNE
AND PRINT THE RESPECTIVE EMPLOYEE DETAILS

```
 package Java8Features;

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

class EmployeeTest
 {
     int empNo;
     String name;
     int age;
     String location;
    public EmployeeTest(int empNo, String name, int age, String location) {
        super();
        this.empNo = empNo;
        this.name = name;
        this.age = age;
        this.location = location;
    }
    @Override
    public String toString() {
        return "EmployeeTest [empNo=" + empNo + ", name=" + name + ", age=" + age + ", location=" + location
                +"]";
    }
 }
public class StreamEmployeeTest {
    public static void main(String[] args) {
        ArrayList<EmployeeTest> al=new ArrayList<EmployeeTest>();
        al.add(new EmployeeTest(1,"Abimanu",21,"mumbai"));
        al.add(new EmployeeTest(2,"Beema",23,"mangalore"));
        al.add(new EmployeeTest(3,"mukesh",22,"chennai"));
        al.add(new EmployeeTest(4,"sakthi",24,"pune"));
        al.add(new EmployeeTest(5,"buvi",25,"pune"));
       ArrayList ls=(ArrayList)al.stream()
        .filter(e->e.location=="pune")
        .collect(Collectors.toList());
        ls.forEach(System.out::println);

    }

}
-----------------------------
OUTPUT:
-------
EmployeeTest [empNo=4, name=sakthi, age=24, location=pune]
EmployeeTest [empNo=5, name=buvi, age=25, location=pune]
----------------------------------
```

3.FILTER THE PASS MARK STUDENT WHOSE MARKS IS 50 AND ABOVE
SOLVE THIS PROBLEM BY USING COUNT AND FILTER METHOD.

```
package Java8Features;

import java.util.ArrayList;
import java.util.stream.Collectors;

class StudentTest
{
    int roll;
    String name;
    int mark;
    public StudentTest(int roll, String name, int mark) {
        super();
        this.roll = roll;
        this.name = name;
        this.mark = mark;
    }

}
public class StreamStudentTest {
    public static void main(String[] args)
    {
        ArrayList<StudentTest> al=new ArrayList<StudentTest>();
        al.add(new StudentTest(1, "mukesh", 95));
        al.add(new StudentTest(2, "logesh", 98));
        al.add(new StudentTest(3, "lite mukesh", 100));
        al.add(new StudentTest(4, "tej", 45));
        al.add(new StudentTest(5, "mehck", 46));
       Long ls= al.stream()
        .filter(s->s.mark>50)
        .collect(Collectors.counting());
       System.out.println(ls);
    }

}
 ----------------------
 OUTPUT :
 --------
 3
 ----------------------------------

20-may-2022       StreamCollect()
***********       ****************

package Java8Features;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Set;
import java.util.stream.Collectors;

class School {
    String section;
    String Depart;
    int roll;
    String name;

    public School(String section, String depart, int roll, String name) {
        super();
        this.section = section;
        Depart = depart;
        this.roll = roll;
        this.name = name;
    }

```

```java
    @Override
    public String toString() {
        return "[section=" + section + ", Depart=" + Depart + ", roll=" + roll + ", name=" + name + " ]";

    }

}

public class CollectorsMethodTest {

    public static void main(String[] args) {
        ArrayList<School> al = new ArrayList<School>();
        al.add(new School("A", "cse", 1, "mukesh"));
        al.add(new School("A", "cse", 2, "basith"));
        al.add(new School("B", "biology", 3, "ajmal"));
        al.add(new School("C", "commerce", 4, "ajmal"));
        System.out.println("         Displaying in List format       ");
        System.out.println("-------------------------------------");
        List<String> lname = al.stream().filter(s -> s.roll < 4).map(s -> s.name.toUpperCase()).collect(Collectors.toList());
        System.out.println(lname);
        System.out.println("-------------------------------------");
        System.out.println("         Displaying in Set format       ");
        System.out.println("-------------------------------------");
        Set<String> s = al.stream().map(st -> st.name.toUpperCase()).collect(Collectors.toSet());
        System.out.println(s);
        System.out.println("-------------------------------------");
        System.out.println("         Joing the String              ");
        System.out.println("---------------------------------------");
        String allNames = al.stream().map(j -> j.name.toUpperCase()).collect(Collectors.joining(" * "));
        System.out.println(allNames);
        System.out.println("---------------------------------------");
        System.out.println("         Grouping                      ");
        System.out.println("---------------------------------------");
        Map<String, List<School>> mapList = al.stream().collect(Collectors.groupingBy(g -> g.section));
        System.out.println(mapList);
        mapList.forEach((k, v) -> System.out.println("Key " + k + "---" + " Value " + v));
        System.out.println("---------------------------------------");
        System.out.println("         Average Finding                ");
        System.out.println("---------------------------------------");
        Double averageOfRoll=al.stream().collect(Collectors.averagingInt(a->a.roll));
        System.out.println("The average is : " +averageOfRoll);
        System.out.println("------------------------------------------");
        System.out.println("          Partision                       ");
        System.out.println("------------------------------------------");
        Map<Boolean, List<School>>mapL=al.stream().collect(Collectors.partitioningBy(p->p.roll>0));
        System.out.println(mapL);
        System.out.println("------------------------------------------");
    }

}
OutPut:
-------

         Displaying in List format
-------------------------------------
[MUKESH, BASITH, AJMAL]
-------------------------------------
         Displaying in Set format
-------------------------------------
[MUKESH, BASITH, AJMAL]
-------------------------------------
         Joing the String
---------------------------------------
MUKESH * BASITH * AJMAL * AJMAL
---------------------------------------
         Grouping
---------------------------------------
{A=[[section=A, Depart=cse, roll=1, name=mukesh ],
[section=A, Depart=cse, roll=2, name=basith ]],
 B=[[section=B, Depart=biology, roll=3, name=ajmal ]],
 C=[[section=C, Depart=commerce, roll=4, name=ajmal ]]}
Key A--- Value [[section=A, Depart=cse, roll=1, name=mukesh ],
 [section=A, Depart=cse, roll=2, name=basith ]]
Key B--- Value [[section=B, Depart=biology, roll=3, name=ajmal ]]
Key C--- Value [[section=C, Depart=commerce, roll=4, name=ajmal ]]
---------------------------------------
         Average Finding
---------------------------------------
The average is : 2.5
------------------------------------------
          Partision
------------------------------------------
{false=[], true=[[section=A, Depart=cse, roll=1, name=mukesh ], [section=A, Depart=cse, roll=2, name=basith ], [section=B, Depart=biology, roll=3, name=ajmal ], [section=C, Depart=commerce, roll=4, name=ajmal ]]}
------------------------------------------
```