Question 1 : Filter using stream API

import java.util.\*;

import java.util.stream.Collectors;

public class FilterAssignment {

    public static void main(String[] args) {

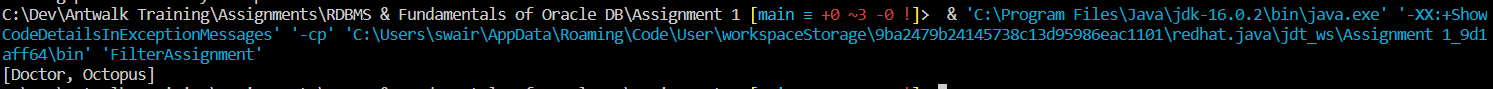
        List<String> l = Arrays.asList("Hello", "Peter", "Doctor", "Octopus", "Big Octopus");

        List<String> ans = l.stream().filter(s -> s.length() > 5 && s.length() < 8).collect(Collectors.toList());

        System.out.println(ans);

    }

}

Output:

Question 2 : Employee – Map Assignment

import java.util.\*;

import java.util.stream.Collectors;

class Employee {

    int id;

    String name;

    double salary;

    public Employee(int id, String name, double salary) {

        super();

        this.id = id;

        this.name = name;

        this.salary = salary;

    }

    @Override

    public String toString() {

        return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";

    }

}

public class MapAssignment {

    public static void main(String[] args) {

        List<Employee> employees = new ArrayList<>();

        employees.add(new Employee(101, "Bruce", 25000));

        employees.add(new Employee(102, "Peter", 15000));

        employees.add(new Employee(103, "Clark", 5000));

        employees.add(new Employee(104, "Lewis", 45000));

        employees.add(new Employee(105, "Sebastian", 55000));

        List<Employee> updatedEmployeeeList = employees

                .stream()

                .map(e -> {

                    e.salary += 0.1 \* e.salary;

                    return e;

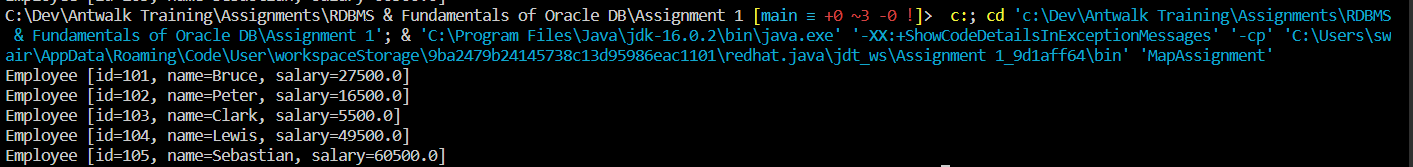
                })

                .collect(Collectors.toList());

        updatedEmployeeeList.stream().forEach(System.out::println);

    }

}

Output :

Question 3 : Product Flatmap Assignment

import java.util.\*;

import java.util.stream.Collectors;

class Products {

    int id;

    String name;

    double price;

    public Products(int id, String name, double price) {

        super();

        this.id = id;

        this.name = name;

        this.price = price;

    }

    @Override

    public String toString() {

        return "product [id=" + id + ", name=" + name + ", price=" + price + "]";

    }

}

public class FlatMapAssignment {

    public static void main(String[] args) {

        List<Products> productList1 = new ArrayList<>();

        List<Products> productList2 = new ArrayList<>();

        List<Products> productList3 = new ArrayList<>();

        List<List<Products>> allProducts = new ArrayList<>();

        productList1.add(new Products(101, "HP Laptop", 35000));

        productList1.add(new Products(102, "Acer Laptop", 65000));

        productList1.add(new Products(103, "Samsung Laptop", 45000));

        productList2.add(new Products(201, "HP Phone", 15999));

        productList2.add(new Products(202, "Acer Phone", 5999));

        productList2.add(new Products(203, "Samsung Phone", 25999));

        productList3.add(new Products(301, "HP Camera", 135000));

        productList3.add(new Products(302, "Acer Camera", 165000));

        productList3.add(new Products(303, "Samsung Camera", 145000));

        allProducts.add(productList1);

        allProducts.add(productList2);

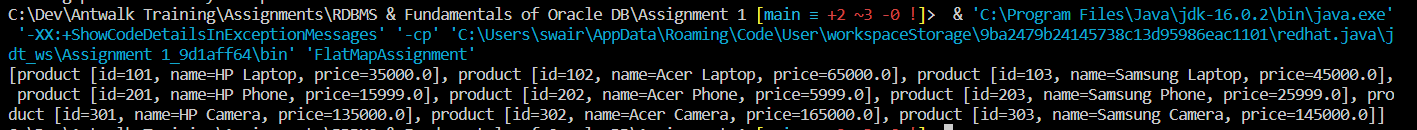
        allProducts.add(productList3);

        List<Products> productsFlatMap= allProducts.stream().flatMap(p -> p.stream()).collect(Collectors.toList());

        System.out.println(productsFlatMap);

    }

}

Output :