* + 1. parameters :

free variables : s, t

* + 1. parameters : s, t

free variables : ignoreCase

* + 1. Supplier<Double> lambda = () -> Math.random();
  1. If two Employee objects have the same name, the return value of compare is 0.

This is now always true because there may be other fields in the Employee class, like salary.

Employee e1 = new Employee(“Win”, 7000);

Employee e2 = new Employee(“Win”, 8000);

As they are having the same name, the compare method from NameComparator will give 0, so e1 and e2 look equal, but they should not be considered equal.

1. Yes, it can.

package prob3;

import java.util.ArrayList;

import java.util.List;

import java.util.function.BiFunction;

public class Main {

public static void main(String[] args) {

BiFunction<Double, Double, List<Double>> bf = (x,y) -> {

List<Double> list = new ArrayList<>();

list.add(Math.pow(x,y));

list.add(x \* y);

return list;

};

System.out.println(bf.apply(2.0, 3.0));

}

}

1. public int countWords(List<String> words, char c, char d, int len) {

List<String> result =words.stream()

.filter(str -> str.length() == len) //checking len equal

.filter(str -> str.indexOf(c) >= 0) //checking c include

.filter(str -> str.indexOf(d) < 0) //checking d not include

.collect(Collectors.toList());

System.out.println(result.toString());

return result.size();

}

* 1. list.forEach(s->System.out.println(s.toUpperCase()));
  2. //list.forEach(String::toUpperCase); //as it’s not printing to console

Function<String, String> f1 = String::toUpperCase;

for(String s : list)

System.out.println(f1.apply(s));

Prob 6 and Lambda and Method Reference Exercises (Example exercises) are in the source code package, also together with the above answers.