**Module-4**

**Assignment-1**

**Code:**

**bikeperformance enum:**

package Module\_4;

public enum BikePerformance {

pulsar(200),yamahaFZ(250),heroExtreme(220),hondaCBR(180);

private int performance;

public int getPerformance() {

return performance;

}

private BikePerformance(int performance) {

this.performance=performance;

}

}

**bikemileage enum:**

package Module\_4;

public enum BikeMileage {

pulsar(50),yamahaFZ(60),heroExtreme(55),hondaCBR(57);

private int mileage;

public int getMileage() {

return mileage;

}

private BikeMileage(int mileage) {

this.mileage=mileage;

}

}

**bikepower enum:**

package Module\_4;

public enum BikePower {

pulsar(40),yamahaFZ(50),heroExtreme(60),hondaCBR(55);

private int power;

public int getPower() {

return power;

}

private BikePower(int power) {

this.power=power;

}

}

**bikeperformancecomparator class:**

package Module\_4;

import java.util.Comparator;

public class BikePerformanceComparator implements Comparator<BikePerformance>{

@Override

public int compare(BikePerformance o1,BikePerformance o2) {

return o1.getPerformance()-o2.getPerformance();

}

}

**bikemileagecomparator class:**

package Module\_4;

import java.util.Comparator;

public class BikeMileageComparator implements Comparator<BikeMileage>{

@Override

public int compare(BikeMileage o1,BikeMileage o2) {

return o1.getMileage()-o2.getMileage();

}

}

**bikepowercomparator class:**

package Module\_4;

import java.util.Comparator;

public class BikePowerComparator implements Comparator<BikePower>{

@Override

public int compare(BikePower o1,BikePower o2) {

return o1.getPower()-o2.getPower();

}

}

**bikemain class:**

package Module\_4;

import java.io.\*;

import java.util.Arrays;

import java.util.List;

import java.util.TreeSet;

public class BikeMain {

public static void main(String args[])throws IOException{

InputStreamReader isr=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(isr);

final List<BikePerformance> list1=Arrays.asList(BikePerformance.values());

TreeSet<BikePerformance> ranks1= new TreeSet<BikePerformance>(new BikePerformanceComparator());

ranks1.addAll(list1);

System.out.println("Bikes in order of increasing performance: ");

for(BikePerformance rank:ranks1) {

System.out.println(rank);

}

final List<BikeMileage> list2=Arrays.asList(BikeMileage.values());

TreeSet<BikeMileage> ranks2=new TreeSet<BikeMileage>(new BikeMileageComparator());

ranks2.addAll(list2);

System.out.println("Bikes in order of increasing mileage: ");

for(BikeMileage rank:ranks2) {

System.out.println(rank);

}

final List<BikePower> list3=Arrays.asList(BikePower.values());

TreeSet<BikePower> ranks3=new TreeSet<BikePower>(new BikePowerComparator());

ranks3.addAll(list3);

System.out.println("Bikes in order of increasing power: ");

for(BikePower rank:ranks3) {

System.out.println(rank);

}

}

}

**Output:**

Bikes in order of increasing performance:

hondaCBR

pulsar

heroExtreme

yamahaFZ

Bikes in order of increasing mileage:

pulsar

heroExtreme

hondaCBR

yamahaFZ

Bikes in order of increasing power:

pulsar

yamahaFZ

hondaCBR

heroExtreme