**Experiment – 3.1**

**Student Name: Vivek UID:**

**Branch: CSE Section/Group:**

**Date of Performance: Semester: 3**

**Subject Name: OOPs using JAVA Subject Code: 21CSH-218**

**Aim of the practical:** Hacker rank problem related to sorting

**Objective:** You are given a list of student information: ID, FirstName, and CGPA. Your task is to rearrange them according to their CGPA in decreasing order. If two students have the same CGPA, then arrange them according to their first name in alphabetical order. If those two students also have the same first name, then order them according to their ID. No two students have the same ID.

Program Code:

#include <iostream>

using namespace std;

int main()

{

cout<<"Hello World";import java.util.\*;

class Student implements Comparator<Student>

{

private

int id;

private

String fname;

private

double cgpa;

public

Student()

{

}

public

Student(int id, String fname, double cgpa)

{

super();

this.id = id;

this.fname = fname;

this.cgpa = cgpa;

}

public

int getId()

{

return id;

}

public

String getFname()

{

return fname;

}

public

double getCgpa()

{

return cgpa;

}

public

int compare(Student a, Student b)

{

if (a.getCgpa() < b.getCgpa())

{

return 1;

}

else if (a.getCgpa() > b.getCgpa())

{

return -1;

}

else

{

if (a.getFname().compareTo(b.getFname()) > 0)

{

return 1;

}

else if (a.getFname().compareTo(b.getFname()) < 0)

{

return -1;

}

else

{

if (a.getId() > b.getId())

{

return 1;

}

else

{

return -1;

}

}

}

}

} public class Solution

{

public

static void main(String[] args)

{

Scanner in = new Scanner(System.in);

int testCases = Integer.parseInt(in.nextLine());

List<Student> studentList = new ArrayList<Student>();

while (testCases > 0)

{

int id = in.nextInt();

String fname = in.next();

double cgpa = in.nextDouble();

Student st = new Student(id, fname, cgpa);

studentList.add(st);

testCases--;

}

Collections.sort(studentList, new Student());

for (Student st : studentList)

{

System.out.println(st.getFname());

}

}

}

return 0;

}

Output:

