Final Exam Pt2 (The Olympics) consists of writing two java files, Person.java and Athlete.java, to track the name of an Athlete and the event that they are competing in at the Olympics. Students are provided with PersonInterface.java that Person.java will implement, AthleteInterface.java that Athlete.java will implement, and an application (driver) – Olympics.java.

Person implements Person Interface and has two constructors. A default no-arg constructor that sets name to "TBD" and one that accepts a String that updates instance field var name

Person //Constructors, instance var (5pts)
- name: String
+ setName(newname: String): void //plain old setter (2pts) + getName(): String //plain old getter (2pts) + toString(): String //returns name instance field (2pts)

Athlete extends Person and implements Athlete Interface. Create a default constructor that sets both name and eventName to "TBD" and a constructor that accepts two Strings and updates inst var name and inst var eventName to the values passed in.

Athlete //Constructors, instance vars (10pts)
- eventName: String - inputFile: String - outputFile: String + ARRAY_SIZE: int //(final) static constant set to 4
+ setFileNames(): void //see below for description (10pts) + csv2Array(): Athlete[] //see below for description (20pts) + setEventName(newEventName:String): void //plain old setter (2pts) + printArray2File(inArray:Athlete[],inMsg:String): void //see below for description (20pts) + compareTo(o:Object): int //provided below + toString(): String //returns <name>, a tab, and then <eventName> (3pts)

Method Descriptions

setFileNames() – collects the location of the inputFile (athletes.csv) and output file and updates the inputFile and outputFile instance vars. Recommend placing athletes.csv in c:/tmp/ (Win OS) and /tmp (Mac OS). Also, recommend using the following path for the output file c:/tmp (Win OS) or /tmp (Mac OS). ***This method should ensure that the user inputs a valid file path for athletes.csv in a while or do while loop.*** (10pts)
csv2Array() – Creates an Athlete [ ] that has ARRAY_SIZE elements. Then reads in a csv file called athletes.csv (via Scanner) whose path is set to inputFile instance field var. An Athlete object should be created for each line in Athletes.csv that sets name and event to the values in the csv file. (20pts) ***athlete.csv is a two(2) column, four(4) row csv file that DOES NOT have a row header.***

HINT – A while loop will be helpful to process each line.

After processing all lines in the Athletes.csv, return the Athlete [ ] to the invoking method in the driver. This method must try catch in the method body and not throws in the message header.

HINT – Scanner will throw a FileNotFoundException if the file is not found.

However, this should not happen since setFileNames( ) ensures a valid path, but the Java compiler still requires a try catch or throws to compile.

setEventName(String newEventName): void //plain old setter (2pts)

printArray2File(Athlete[] inArray, String inMsg) - writes inMsg, followed by output header, followed by contents of Athlete[ ] produced in csv2Array to a text file (via PrintWriter) in nice columns (see below for final output) (20pts). NOTE – Initially, the columns might not line up. However, if you print to a printer or pdf, it will be lined up.
HINT – Use of "\t" for the header and the .toString( ), will help to line up your columns.

compareTo(Object o) – Copy and paste below into Athlete.java

```
/**
     * CompareTo to be used in sorting Athlete[]'s
     *
     * @param o Object that is passed to determine if it is less than,
     * equal, or more than calling object.
     * @return -1,0,1
     */
    @Override
    public int compareTo(Object o) {
        if (o != null && this.getClass() == o.getClass()) {
            Athlete temp = (Athlete) o;
            return this.eventName.compareTo(temp.eventName);
        }
        return -1;
    }
```

```
***With TBDs***
Athlete          Event Name
Sam Swimmer      Swimming
Joe Boxer        Boxing
Harry Harpoon    TBD
TBD              400m Sprint


***Before sorting events***
Athlete          Event Name
Sam Swimmer      Swimming
Joe Boxer        Boxing
Harry Harpoon    Javelin
Tommy Track      400m Sprint


***After sorting events***
Athlete          Event Name
Tommy Track      400m Sprint
```

```
Joe Boxer        Boxing
Harry Harpoon    Javelin
Sam Swimmer      Swimming
```

Item	Pts	Pts
Person - Constructors (1+3=4pts), instance var (1pt)	5	
setName(String newName): void //plain old setter	2	
getName(): String //plain old getter	2	
toString(): String //Returns name instance field	2	
Athlete - Constructors (2+4=6pts), instance vars (3pts), ARRAY_SIZE constant	10	
setFileNames(): void - while pit (5pts), everything else (5pts)	10	
setEventName(String newEventName): void //plain old setter	2	
csv2Array(): Athlete[]	16	
printArray2File(Athlete[] inArray, String inMsg): void	16	
toString(): String //Prints out <name>, a tab, and then <event name>	3	
Constructor javadocs-Person	2	
Constructor javadocs-Athlete	2	
Method javadocs-Person	4	
Method javadocs-Athlete	4	
	80	0