Contestant Number: _____

Time: _____

Rank: _____

JAVA PROGRAMMING
(340)

STATE – 2019

Production Portion:

Program 1: Team Directory _____ (305 points)

TOTAL POINTS _____ ***(305 points)***

Failure to adhere to any of the following rules will result in disqualification:

1. **Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
2. **No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
3. **Electronic devices will be monitored according to ACT standards.**

No more than ten (10) minutes orientation
No more than ninety (90) minutes testing time
No more than ten (10) minutes wrap-up

You will have ninety (90) minutes to complete your work.

Your name and/or school name should *not* appear on work you submit for grading.

1. Create a folder on the flash drive provided using your contestant number as the name of the folder.
2. Copy your entire solution/project into this folder.
3. Submit your entire solution/project so that the graders may open your project to review the source code.
4. Ensure that the files required to run your program are present and will execute on the flash drive provided.

   *Note that the flash drive letter may *not* be the same when the program is graded as it was when you created the program.

   * It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter.

   The graders will *not* compile or alter your source code to correct for this.
   Submissions that do *not* contain source code will *not* be graded.

Assumptions to make when taking this assessment:

- The input file will contain only ASCII characters.
- A test input file will be available and named, "team.txt."

Development Standards:

- Your Code must use a consistent variable naming convention.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).
- If you create a class, then you must use Javadoc comments.

Team Directory

Create a program that implements an insertion sort for a directory of team members. You will first create a simple TeamMember class to represent each member of the team.

TeamMember

Variables:

- String fullName – Holds the person's full name
- String idString – Holds a String representing the person's unique team ID

Methods:

- TeamMember(String name, String id) – Constructor that takes in a String parameter representing the person's name, and a String parameter for the team member's ID. Names should be converted to title case.
- String toString() – Returns the name of the team member.
- Int compareTo(TeamMember other) – Compares the idString of this TeamMember to the idString of other. Use the String.compareTo() method to compare the two Strings and return a -1, 0, or 1.

In a separate Sort class, you are required to fill in a main method from the input file. You can write additional helper methods if needed.

- The file will consist of "STOP", in any combination of lowercase and uppercase letters, to end the file.
- After the file is read, sort the ArrayList in increasing order by ID using the insertion sort algorithm.
- After the file is read, print the contents of the sorted ArrayList using ArrayList.toString().

Input:

The input to this problem consists of a text file. The first line of the text file contains the team member name. The next line will consist of the team member id.

Output:

The output consists of one line of text for the sorted ArrayList.

Example input:
zeb
02-003
rita stevens
01-001
SUE wOOds
02-001
adele
01-002
BarBara
02-002
STOP

Example output:

[Rita Stevens, Adele, Sue Woods, Barbara, Zeb]

Requirements:

1. You must create an application with the main class named *Sort.*
2. You must create an additional class within the application named *TeamMember.*
3. Your contestant number must appear as a comment at the top of the main source code file.
4. If the input file is not found, then the program should display an appropriate message and exit.
5. The program will perform the required tasks correctly for however many team members are indicated in the input file.
6. The program must implement methods to:
   a. Constructor that takes in a String parameter representing the person's name, and a String parameter for the team member's ID.
   b. Return the name of the team member.
   c. Compare the idString of this TeamMember to the idString of another.
   d. Print the contents of the sorted array.
7. The program will display the output like the example above for all of the values in the input file.
8. Do not use any of the methods provided by the class java.util.Collections.

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive	_____	10 points
The projects main class is named **Sort**	_____	10 points
The projects helper class is named **TeamMember**	_____	10 points

Program Execution

The program runs from the USB flash drive	____	15 points

If the program does *not* execute, then the remaining items in this section receive a score of zero.

The program runs and reads the input file	____	30 points
The program displays an error message if the file cannot be found	____	20 points
The program converts the input to title case	____	20 points
The program sorts the input into a new ArrayList, in correct order	____	30 points
The program stops reading the input file when "STOP" is found	____	20 points
The program displays the sorted list, in correct order	____	20 points

Source Code Review

The source code is properly commented		
A comment containing the contestant number is present	____	10 points
Methods and code sections are commented	____	20 points
A method exists to perform the insert into the ArrayList	____	20 points
A method exists to perform the compareTo	____	20 points
A method exists to perform the display of output	____	20 points
Code uses try... catch for exception handling	____	20 points
Code uses a consistent variable naming convention	____	10 points

Total Points = 305