CSC 103 Programming Assignment #4 10/15/16

Due date – **Friday December 2nd @ Noon**

This project is to be done by a pair (2) of students. No groups of three or one without the instructor’s permission. The phrase “I like to work alone” is not an acceptable reason for not having a partner. You will need to learn to work together. Make sure that both of you understand all aspects of the assignment. Note: If the work is not evenly done, this will no doubt be seen during exams.

**Project # 4 Golfer Scores Database using a Binary Search Tree**

You are to create a Program that simulates a Database with Golfer information that is stored in a Binary Search Tree.

Summary of program:

Your program will start with reading an input file filled with golfer information. Each line of this file will contain the needed information to create a golfer object. This file will be given to you. As each line of the file is read, you will create a Golfer object and place it in the Binary Search Tree. Then a menu will be displayed giving you choices to display golfer information, update, remove, or add new golfer. When quit is chosen the contents of your binary search tree will be traversed and output to update the same datafile. (You will be writing over the original data) You may want to create a file of a different name for testing purposes, so that you do not corrupt the original file during debugging.

**Golfer** class:

The **Golfer** class will contain the following private data variables:

-lastname

-numberOfRounds

-averageScore (double)

-handicap (positive integer value between 0- 20)

Must also contain the following methods:

-constructor (maybe two)

-accessors for each of the four variables

-mutators for each of the four variables

-method to add a new score (updates numberOfRounds and averageScore)

-**compareTo** method to help sort (class will implement Comparable)

-**toString** that will convert to one string all of the stored info

You may wish to add other methods to the Golfer class for your own use. I am keeping it simple for this project, but someone could add information on calculating handicap values, or keep track of other statistics such as number of putts or what is par for the course.

To create the Binary Search Tree you will be using the **BTNode<E>** class given to you and you will be finishing methods in the template class **TreeBag**.

In the **TreeBag** class you only need to implement methods that you will be using for this project. You are not required to implement the rest of the methods in the class. You may leave them as program stubs and ignore. Make sure all references to Golfer objects are generic ones. I have written Golfer Object in as the parameters so that you better understand what each method is doing, but in the actual Treebag class they should be listed as “E”. Among the methods that you *will* need to create are:

* **public void add (Golfer Object)**

You will be adding golfers in the tree to locations based on the golfer’s lastname, but there is method in the Golfer class that you have created already called **compareTo**. Therefore you should use the **compareTo** method to compare this new golfer to golfers already within the tree.

* **public GolferObject retrieve (Golfer Object).**

The **retrieve** method will search the Binary Search tree for the element, the method will return the object that was found. If the object is not found the method returns null. You will need to create a Golfer object with only the lastname, in order to do the search. I created an extra constructor for this task. When **compareTo** gives a zero, you know that you have found the correct Node.

* **public boolean remove (Golfer Object)**

The **remove** method returns boolean therefore you need first find if the object is in the tree, if not return false, if yes, then delete the node from the tree and return true. Make careful sure that after Golfer removal you still have a valid binary search tree. (Handle all cases)

* **public void display()**

Display the golfer’s in alphabetica**l** order based on the lastname (think of the appropriate traversal through the tree to get this)

You may want to create other methods that you will be using in the **TreeBag** class.

You are to make use of the following class that we have discussed without making any changes. These classes are stored in the project folder.

* **BTNode<E>** (unaltered)

You will create these classes to finish the project:

* **Golfer** (explained above)
* **TreeBag** (template given)
* **GolferScoresTree**- main driver of program.

The main driver was described above and contains the main menu which must contain the following choices:

Menu choices:

1. Display listing to screen of all golfers stats(ordered by lastname)
2. Display the golfers in current tree format(Use displayAsTree )
3. Find and display one individual golfers stats
4. Update an individual golfers stats(by adding an additional score)
5. Remove a golfer from the Database
6. Add a new golfer to the Database
7. Quit and update datafile

The Datafile used is data stored in text format (not binary, so that we can read it) with one golfer on each line and spaces between fields. It is organized like this:

**lastname numberOfRounds handicap average**

The sample data file given is called **golferinfo.txt** and will be used as a starting point to test your program. A copy of this file is given in with these program directions.

Tips for good grades:

* Make sure you use comments where needed and use variable names that make sense, some of your grade will depend on program style as well as the use of your program.
* Update the comments in the class file, to include your names and any new information
* You will lose points for things like not indenting, or naming variables in non-descriptive ways. Do no leave in debugging code, or commented out code.
* I use jGrasp and the java version that is in the lab computers. So make sure that your programs work with this.
* Test your own projects thoroughly before you hand them in.
* Late projects will not be accepted so plan ahead.

The four classes you are using for this project should be in separate files. Name them **GolferScoresTree.java, Golfer.java , TreeBag.java** and **BTNode.java**. If you do not name these files correctly, you will lose points. Use Javadoc to create the documentation for your **TreeBag** class.

Hand in electronically – (NOT E-mail!!!)

In S-drive CSC 103 folder:

1. Create folder called **projectfour\_firstname\_lastname**
2. Place these four script files named above in folder.
3. Place any other class files that you created.
4. Place any other documentation such as a readme file in here as well.