GEEN165 Major Programming Assignment 1 - Fall 2016
Due: September 30, 2016 @ 9pm

Introduction:

You are required to create an application to help manage a golf league. Each team will contain a team name and five golfers. Each week, teams golf head-to-head and submit the results to your application which summarizes the results.

Golfer Class:

Golfer	
-golferID:int	Unique number for each golfer
-firstName:String	Golfer first name
-lastName:String	Golfer last name
-totalRounds:int	Total number of rounds of golf
-totalStrokes:int	Sum of scores from all rounds
+Golfer()	Set all values to Java defaults
+Golfer(golferID:int, firstName:string …)	Constructor with parameter for each data field.
+//mutators and accessors	
+getAverage():double	Return the golfer's average (totalStrokes/totalRounds)
+toString():String	Return the golfer's properties

GolfTeam Class:

GolfingTeam	
-teamID:int	Unique number for each team
-teamName:String	Team name
-golfers:Golfer[5]	The golfers on the team
-roundsWon:int	Total rounds won
-roundsLost:int	Total rounds lost
+GolfTeam()	Set all values to Java defaults
+GolfTeam(teamID:int, teamName:String)	Constructor with teamID and teamName parameters
+//mutators and accessors	
+getGolfer(index:int):Golfer	Return the Golfer at position index.
+setGolfer(index:int, golfer:Golfer)	Set the value to bower @ position index
+getAverage():double	Return the Team's average (totalStrokes/totalRounds)
+toString():String	Return a string with all Team Properties

Golf League Class:

League	
-leagueName:String -numTeams:int -teams:GolfTeam[] -numWeeks	Name of the league Number of teams in the league All of the golfing teams Number of weeks teams have golfed
+League() +League(leagueName:String, numTeams:int, numWeeks:int) +readWeeklyData(wklyFileName:String):void +writeLeagueData(configFileName:String):void +getLeagueStandingsReport() : String +//mutators and accessors +getTeam(index:int):Team +setTeam(index:int, Team team) +toString():String	Set all values to Java defaults Constructor with parameter for each data field. Read and aggregate weekly golfing data. Write league data in prescribed input format Return a league standings report. Return Team at position index. Set Team at position index to team. Return all the League properties.

Note: The format of the toString() method for all classes should match the format of the League input file. This will allow you to use the toString() method to greatly simplify the writeLeagueData() method.

Input file formats:

Your program will read in the League Input file when it first starts executing. Users will have the option of loading weekly files which should update each golfer's statistics.

League Input File format:
League Name
NumTeams NumWeeks
Team0_ID Team0_wins Team0_losses Team0_Name
Golfer0 ID
Golfer0 fname
Golfer0 lname
Golfer0_totalRounds Golfer0_totalStrokes
*** Repeat for other 4 Golfers ***
*** Repeat for other Teams ***

League Standings Report:

The header for the report should include the league name and the total number of weeks the teams have golfed. Next, the League Standings Report should display a list of all teams sorted in decreasing order of number of wins (i.e. the team with the most wins goes at the top). Each line of output should have the team rank, the team number, the team name, number of wins and number of losses. Finally, the report should display information about each team in **sorted order by team number**. The information for each team should include the team name, the team number, team wins, team losses and information about each golfer on the team (golfer name, golfer id, total rounds played and average strokes per round).

Output File:

Once a weekly file is aggregated with the existing league data, the user should have the option of saving the aggregated league information into an output file that matches the format of the League input file format. This will enable the aggregated data to be loaded into the program the next time it is executed. Your program should prompt the user for the name of the desired output file name.

Level 0 - A program that does not compile receives a grade of 0 (zero).

Level 1 Assignment (Max Grade=20%):

Implement the Golfer class. Create a main the tests both constructors and all the other methods in the class. Use the toString() method to show the contents of the Golfer objects.

Weekly File format:
TeamID rounds_won rounds_lost
GolferID
Round1 score
Round2 score
*** Repeat for other golfers on team that golfed ***
*** Repeat for other teams ***

Level 2 Assignment (Max Grade=40%):

Add the GolfTeam class to your assignment. Create a main method that will instantiate 5 golfers and add them to a GolfTeam object. Demonstrate correctness using the toString() method of the GolfTeam class.

Level 3 Assignment (Max Grade=60%):

Add the League class to your assignment. Your *main* method must be able to populate the League object using a League input file. Use command-line arguments to supply the League data file name to your main() method**.** The following is a brief algorithm for the method *readLeagueData* that reads your League data. This method should be implemented in your same class as your *main* method. You need to pass args[0] as a parameter (A good developer would check the length of the args array to make sure a file name was provided -- if not, use the JFileChooser to prompt the user for a file name.)**. Here is a simple algorithm for that method:**

+readLeagueData(leagueFileName:String): League

- Create Scanner object using League data file name.
-Read in the League name, number of teams and weeks golfed.
-Create a League object now that you know the number of teams.
-For each of the N teams (where N represents the number of teams read in above):
    -Create/instantiate a new Team object
    - Read the teamID, team name, numWins, numLosses
    -For each of the five golfers on the team:
        - Create a new Golfer object and read in the Golfer information.
        -Add the golfer to the array property of the team object
    -Add the Team object to the League object.
-Return the League object as the return type of the function. Note: You should save the return value to a League variable declared in your main().

You do not have to implement the readWeeklyData and writeLeagueData methods at this assignment level.

Level 4 Assignment (Max Grade = 80%):

Your program will implement the readWeeklyData and writeLeagueData methods of the League class. Prompt the user for the respective filenames using the JFileChooser. The readWeeklyData file is used to update the data for the golfers and team. The rounds won and rounds lost integer values are used to update the total strokes for the respective golfer. The total rounds for each golfer is updated by 3 each week (assume each golfer golfs each week). Teams can win or lose up to 4 rounds/points each week (the three rounds golfed plus the team that has the most total strokes for the three rounds gets the 4th point). After the weekly data is read and incorporated into the golfer and team statistics, use the JFileChooser to allow the user to save the updated league data into a file with the same format as the League data file. You will need to sort the Teams based off of the roundsWon property to update the team rank each week.

Level 5 Assignment (Max Grade = 100%):

*** An addendum to this assignment will be posted with details of this assignment level. It will involve using the NetBeans GUI builder to provide an interface for your application.

Documentation:

Look over the coding style guide for the documentation standard. Uncommented programs will be penalized heavily so I encourage you to comment as you go.

Submisssion:

Use an archival tool to zip your entire NetBeans Project. Create a file named README.txt that describes what level you think you completed. Use the upload link for the assignment to submit your zipped project. Please respect the class honor code and do your own work. Start early and have fun.